

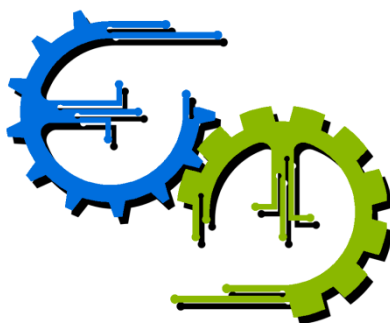


TRABALHO DE GRADUAÇÃO

**ANÁLISE DE DESEMPENHO E
CONTROLE DE UM ROBÔ SCARA**

Por,
Edrysson Sebastien Araujo Rocha

Brasília, Dezembro de 2016



**ENGENHARIA
MECATRÔNICA**

UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA

Faculdade de Tecnologia

Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

**ANÁLISE DE DESEMPENHO E
CONTROLE DE UM ROBÔ SCARA**

POR,

Edrysson Sebastien Araujo Rocha

Relatório submetido como requisito parcial para obtenção
do grau de Engenheiro de Controle e Automação.

Banca Examinadora

Prof. Walter de Britto Vidal Filho, UnB/ ENM
(Orientador)

Guilherme Caribé de Carvalho, UnB/ ENM

Prof. José Maurício Santos T. da Motta, UnB/
ENM

Brasília, Dezembro de 2016

FICHA CATALOGRÁFICA

EDRYSSON, ROCHA

Análise de desempenho e controle de um robô SCARA,

[Distrito Federal] 2016.

xvii, 57p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2016). Trabalho de Graduação – Universidade de Brasília. Faculdade de Tecnologia.

1. Modelo

2. Controle

3. Geração de trajetória

4. Análise de desempenho

I. Mecatrônica/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

ROCHA, E., 2016. Análise de desempenho e controle de um robô SCARA. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-nº 31/2016, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 123p.

CESSÃO DE DIREITOS

AUTOR: Edrysson Sebastien Araujo Rocha.

ANÁLISE DE DESEMPENHO E CONTROLE DE UM ROBÔ SCARA: Análise de desempenho, modelagem, métodos de controle, desenvolvimento de programa para um robô SCARA.

GRAU: Engenheiro

ANO: 2016

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

Edrysson Sebastien Araujo Rocha

Sqn 114N Bloco G apto 310–Asa Norte.

70764-070 Brasília – DF – Brasil.

AGRADECIMENTOS

Agradeço primeiramente a Deus, pois reconheço ele como fonte de inspiração e de vida de onde vem toda minha motivação para ser relevante na vida pessoal, profissional e acadêmica. Agradeço a minha família pelo apoio, ao professor Walter de Britto pelo acompanhamento no desenvolvimento deste projeto, ao João Carlos com quem eu dividi a bancada de trabalho no graco e me deu diversas dicas, ao Claudio (Pereira) técnico do graco que me ajudou nas necessidades mecânicas do projeto e a todos os meus companheiros de curso.

Edrysson Sebastien Araujo Rocha.

RESUMO

Este trabalho é desenvolvido sobre um robô didático de topologia SCARA. É realizada a análise de desempenho do robô, medição dos desvios de projeto, o levantamento de modelos e comparação de modelos. Após a modelagem estudam-se dois métodos de resolver o problema da cinemática inversa, estes são, Denavit-Hartenberg e Jacobiano inverso, está ainda no escopo desse trabalho algoritmos simples para seguir trajetória. É desenvolvido também um projeto eletrônico para o controle dos motores de passo do robô e um projeto de software para fornecer uma interface de controle ao usuário do robô.

Palavras Chave: Modelagem, Controle, Denavit-Hartenberg, Jacobiano, Trajetória.

ABSTRACT

This work is developed on a SCARA topology didactic robot. It performs the analysis of robot performance, measurement of design deviations, model survey and model comparison. After the modeling is studied two methods of solving the problem of the inverse kinematics, these are, Denavit-Hartenberg and inverse Jacobian, It is still within the scope of this work simple algorithms to follow the trajectory. An electronic design for the control of robot step motors and a software project to provide a user control interface of the robot is also developed.

Keywords: Modeling; Control; Denavit-Hartenber; Jacobia; Trajectory;

SUMÁRIO

1	INTRODUÇÃO	1
1.1	DESCRIÇÃO DO OBJETIVO.....	1
1.2	ORGANIZAÇÃO DO TRABALHO	2
2	REVISÃO BIBLIOGRÁFICA	4
2.1	ROBÔ	4
2.2	MOTOR DE PASSO	4
2.2.1	PONTE-H	5
2.3	DENAVIT-HARTENBERG	5
3	MEDIÇÃO GEOMÉTRICA.....	7
3.1	BRAÇO DE MEDIÇÃO	8
3.2	COLETA DE DADOS.....	9
3.2.1	BASE.....	10
3.2.2	HASTE DE SUSTENTAÇÃO	11
3.2.3	BASE DOS ELOS.....	12
3.2.4	EIXO JUNTAS	13
3.3	ANÁLISE DE DADOS.....	14
3.3.1	RELAÇÃO ENTRE PLANOS	14
3.3.2	RELAÇÃO ENTRE PLANOS E RETAS	15
3.3.3	RELAÇÃO ENTRE RETAS.....	16
4	MODELOS CINEMÁTICOS.....	17
4.1	CINEMÁTICA DIRETA.....	17
4.1.1	MODELO PROJETADO	17
4.1.2	MODELO MELHORADO	18
4.1.3	MODELO REAL.....	18
4.2	COMPARAÇÕES	19
4.2.1	JUNTAS DE PLANO.....	19

4.2.2	ANALISE DE RESULTADOS DAS JUNTAS DE PLANO	22
4.3	JUNTA VERTICAL.....	23
5	MÉTODOS DE CONTROLE CINEMÁTICO	25
5.1	MÉTODO DE PREVISÃO DE CORREÇÃO EM Z	25
5.2	MÉTODO DE TRAJETÓRIA POR JACOBIANO INVERSO	27
5.3	MÉTODO HÍBRIDO	29
6	GERAÇÃO DE TRAJETÓRIA.....	31
6.1	INTERPOLAÇÃO.....	31
6.2	JACOBIANO INVERSO	31
6.3	CINEMÁTICA INVERSA	31
6.4	CRITÉRIOS DA ESCOLHA	32
7	PROJETO E IMPLEMENTAÇÃO ELETRÔNICA	33
7.1	MOTORES	33
7.2	DRIVERS DE POTÊNCIA.....	33
7.2.1	L298N.....	34
7.3	TTL 7404	34
7.4	CONTROLE.....	35
7.5	FONTE	35
7.6	IMPLEMENTAÇÃO ELETRONICA	36
8	PROJETO E IMPLEMENTAÇÃO DE SOFTWARE.....	39
8.1	MÓDULO ARDUINO.....	39
8.2	MÓDULO COMPUTADOR	40
8.2.1	INTERFACE GRÁFICA.....	40
8.2.2	OPERAÇÃO DO ROBÔ.....	41
8.3	CONTROLE POR CÓDIGO.....	41
8.3.1	MODEM/MODEC.....	43
8.3.2	MOV	43
8.3.3	MOVR.....	43

8.3.4	MOV P.....	44
8.4	CONTROLE.....	45
9	ANÁLISE DE DESEMPENHO.....	46
9.1	Teste de desempenho	47
9.1.1	REPETIBILIDADE DE POSICIONAMENTO	47
9.1.2	ACURACIDADE DE POSICIONAMENTO	49
9.1.3	REPETIBILIDADE DE TRAJETÓRIA.....	49
10	CONCLUSÃO.....	51
10.1	PROJETOS FUTUROS	51
11	REFERÊNCIAS BIBLIOGRÁFICAS.....	53

LISTA DE FIGURAS

Figura 1 - Esquemático L298N	5
Figura 2 - exemplo de cálculo dos parâmetros DH	6
Figura 3 – Robô SCARA.....	7
Figura 4 - Braço de medição.....	9
Figura 5 - Ilustração do plano de base obtido através do braço de medição.	11
Figura 6 - Ilustração de um dos planos da haste de sustentação obtido através do braço de medição.	12
Figura 7 - Nomenclatura de algumas faces de estruturas do robô.	13
Figura 8 - Ilustração do eixo de uma junta obtido através do braço de medição.	14
Figura 9 - Modelo projetado x modelo real (mm).....	20
Figura 10 - Modelo projetado x modelo melhorado (mm).	21
Figura 11 - Modelo melhorado x modelo real (mm).	21
Figura 12 - Comparação entre modelos (mm).....	22
Figura 13 - Driver de potência baseado em L298N	34
Figura 14 - Pinagem TTL 7404	35
Figura 15 - Fluxografo parte eletrônica.	36
Figura 16 – Protoshield	36
Figura 17 - Sistema eletrônico com caixa protetora aberta.	37
Figura 18 - Sistema eletrônico com caixa protetora fechada.	38
Figura 19 – Conjunto arduino e protoshield conectado ao sistema.	38
Figura 20 - Funcionamento arduino.	40
Figura 21 - modo coordenadas	41
Figura 22 - modo código.	42
Figura 23 - Funcionamento PC.	42
Figura 24 - Exemplo de reta com movr.	44
Figura 25 - Exemplo de parabola com comando movp.	45
Figura 26 - Acuracidade e Repetibilidade	46

Figura 27 - marcação PT1.	48
Figura 28 - marcação PT2.	48
Figura 29 - marcação PT3.	48
Figura 30 - marcação PT4.	48
Figura 31 - marcação PT5.	49
Figura 32 - Resultado teste repetibilidade de trajetória.	50

LISTA DE TABELAS

Tabela 1 - Parâmetros de projeto do robô.	2
Tabela 2 - Características do Braço de Medição.	8
Tabela 3 - Dados plano base.....	10
Tabela 4 - Relações entre planos.....	15
Tabela 5 - Relações entre retas e planos.	15
Tabela 6 - Ângulo entre retas.	16
Tabela 7 - Distância entre retas (mm).	16
Tabela 8 - Modelo projetado x modelo real (mm).	20
Tabela 9 - Modelo projetado x modelo melhorado (mm).....	20
Tabela 10 - Modelo melhorado x modelo real (mm).....	21
Tabela 11 - Comparação entre modelos (mm).	22
Tabela 12- Erro malha intermediária sem quantização.....	27
Tabela 13- Erro malha intermediária com quantização.....	27
Tabela 14 - Erro malha de teste sem quantização com método híbrido.....	29
Tabela 15 – Erro malha de teste com quantização com método híbrido.....	29
Tabela 16 - Erro malha de teste com quantização com cálculo real da cinemática inversa. .	30
Tabela 17 - Conexões arduino.	37
Tabela 18 - Conexões lógicas l298N.	37
Tabela 19 - Lista de comandos.	43
Tabela 20 – Pontos de teste (mm).....	47
Tabela 21 - Ordem dos pontos de teste por etapa.....	47
Tabela 22 - Resultado medição pontos de teste (mm).....	49
Tabela 23 - Erro absoluto dos pontos de teste (mm).	49
Tabela 24 - Reta de teste (mm).....	50

LISTA DE SÍMBOLOS

Símbolos Latinos

l_n Distância entre junta n e n+1. [mm]

Símbolos Gregos

θ_n Ângulo junta n [rad]

Siglas

DH Denavit-Hartenberg

PWM Pulse Width Modulation

SDL Simple Direct Layer

1 INTRODUÇÃO

Projetos envolvendo robôs são de interesse especial para o curso de engenharia mecatrônica pois permitem um estudo e desenvolvimento interdisciplinar envolvendo as áreas do curso: mecânica, eletrônica e computação. Com a falta de fornecedores de robôs de baixo custo torna-se uma opção interessante o desenvolvimento de robôs didático. Os robôs didáticos de baixo custo, desenvolvidos sem equipamentos de alta precisão tem muitos desvios de projeto, isto é, a estrutura do robô não sai conforme projetado. É então necessário fazer uma análise desses desvios de projetos, um levantamento de seus efeitos no desempenho do robô e então propor uma correção desses efeitos, como a correção das estruturas mecânicas do robô tem custo elevado é interessante que essa correção seja feita por software.

1.1 DESCRIÇÃO DO OBJETIVO

O robô analisado neste trabalho foi desenvolvido por [1] em um projeto de graduação anterior. A partir do projeto descrito aqui nos próximos capítulos serão feitas análises da conformidade do robô construído com o projeto, bem como o desempenho do robô. A Tabela 1 define parâmetros básicos do robô.

Com os dados do projeto levantados é possível se trabalhar no modelo do robô e obter as características relevantes para análise de seu desempenho. A partir dos dados é possível estudar o modelo cinemático do manipulador, sendo de interesse para análise a comparação dos dados de projeto com os dados obtidos por medição deste. Além de útil para a análise, o modelo cinemático também é necessário para implementar a estratégia de controle cinemático da posição do efetuador terminal em função das posições de juntas. Apesar de existirem algoritmos de trajetória que independem do algoritmo de posição como o que usa a inversa do jacobiano (que é estudado nesse trabalho) existem também algoritmos que usam o algoritmo de posicionamento, por exemplo, fazendo uma interpolação da trajetória (também estudado nesse trabalho), Assim, é importante para essa classe de algoritmos se ter um bom algoritmo de posição, e no caso do algoritmo por jacobiano inverso é importante um bom modelo visto que é a partir do modelo que se calcula o jacobiano.

TOPOLOGIA	Scara
GRAUS DE LIBERDADE	3
ALCANCE PLANO	201.91mm – 280mm
ALCANCE VERTICAL	0 – 100mm
JUNTA 1	Revolução
JUNTA 2	Revolução
JUNTA 3	Prismática
ALCANCE JUNTA 1	0° – 180°
ALCANCE JUNTA 2	-90° – 90°
ALCANCE JUNTA 3	0mm – 100mm
DISTÂNCIA JUNTAS 1 E 2	168mm
DISTÂNCIA JUNTAS 2 E 3	112mm
RESOLUÇÃO JUNTA 1	1.8°
RESOLUÇÃO JUNTA 2	1.8°
RESOLUÇÃO JUNTA 3	0.006mm

Tabela 1 - Parâmetros de projeto do robô.

O objetivo geral é analisar o desempenho do robô desenvolvido em [1] e propor a correção de seus desvios de construção. E tem como objetivos específicos:

- Medição geométrica do robô construído para identificação dos desvios de projeto;
- A modelagem cinemática e correção do modelo;
- Implementação em software do controle cinemático;
- Implementação em software de controle cinemático com geração de trajetória.

1.2 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado da seguinte forma:

- **REVISÃO BIBLIOGRÁFICA:** Serão tratados os conceitos necessário para entender o trabalho.
- **MEDIÇÃO GEOMÉTRICA:** processo de medição das estruturas geométricas do robô.

- **MODELOS CINEMÁTICOS:** Estudo comparativo de três modelos cinemáticos.
- **MÉTODOS DE CONTROLE CINEMÁTICO:** Estudo de dois métodos de controle cinemático e de um método híbrido entre os dois.
- **GERAÇÃO DE TRAJETÓRIA:** Uma análise breve de dois métodos de geração de trajetória e a seleção de um.
- **PROJETO E IMPLEMENTAÇÃO ELETRÔNICA:** Seleção de placa de controle, placas de potência e dispositivos lógico e montagem da eletrônica.
- **PROJETO E IMPLEMENTAÇÃO DE SOFTWARE:** Capacidades e funcionamento do software de controle implementado.
- **ANÁLISE DE DESEMPENHO:** Análise de desempenho do manipulador aplicando-se os itens estudados neste projeto.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo trata de conceitos necessários para o entendimento deste trabalho. É introduzido, por exemplo, o robô utilizado neste trabalho, bem como conceitos de modelagem e desempenho.

2.1 ROBÔ

Antes dos robôs de aplicações práticas surgirem, o termo robô já existia na ficção, portanto, é um conceito meio abstrato. Existem várias definições de robôs sendo importante ressaltar as da JIRA (Japanese Industrial Robot Association) e da RIA (Robot of Institute of America) (citadas em [1]). A definição da JIRA é mais abrangente, enquanto a da RIA é mais restritiva e aplicada a este trabalho.

A RIA define robô como um manipulador reprogramável, multifuncional projetado para mover material, partes, ferramentas, ou peças especializadas através de um algoritmo reprogramável para executar uma variedade de tarefas.

Os robôs manipuladores são de maneira geral composto por juntas e elos. As juntas possibilitam a movimentação sendo tipicamente de revolução (para movimentos rotacionais) ou prismáticas (para movimentos lineares) [1]. Os manipuladores podem ser classificados pelos seus tipos de juntas e sua geometria.

O manipulador a ser tratado nesse trabalho é do tipo SCARA. O manipulador do tipo SCARA possui duas ou três juntas de revoluções coplanares que permitem o movimento horizontal do manipulador e uma junta prismática para o movimento vertical [1].

Além do trabalho [1] do qual este é continuação existem outros projetos de desenvolvimento deste tipo de manipulador. Na PUC Minas [2] foi desenvolvido foi desenvolvido um SCARA com quatro juntas sendo três verticais (uma das juntas é apenas utilizada para orientação do efetuador terminal) e uma planar este robô. Em [3] é descrito o projeto de um robô SCARA com cinco graus de liberdade o qual possui uma garra como efetuador terminal.

2.2 MOTOR DE PASSO

O motor de passo é um motor DC sem escova utilizado para posicionamento de precisão. O motor de passo divide uma volta completa em vários passos sendo possível

comandar o motor para parar em um passo específico. Contando que se respeite a velocidade de comutação e o torque do motor não é necessário nenhum sensor de feedback para seu controle, sendo assim possível o controle em malha aberta [4].

2.2.1 PONTE-H

A ponte H é um circuito que permite que a corrente circule nos dois sentidos entre seus terminais com elas pode-se se acionar um motor de passo de forma bipolar, ou seja, a mesma bobina ao ser chaveada é energizada num sentido contrário. O acionamento bipolar exige um driver de potência mais complexo como, por exemplo, uma ponte H, porém permite que o motor trabalhe com mais torque, pois energiza toda a bobina diferente do acionamento unipolar que energiza meia bobina. Por essas razões a ponte H se torna um circuito importante para esse trabalho. A Figura 1 mostra o esquemático do driver usado nesse trabalho que tem duas pontes H.

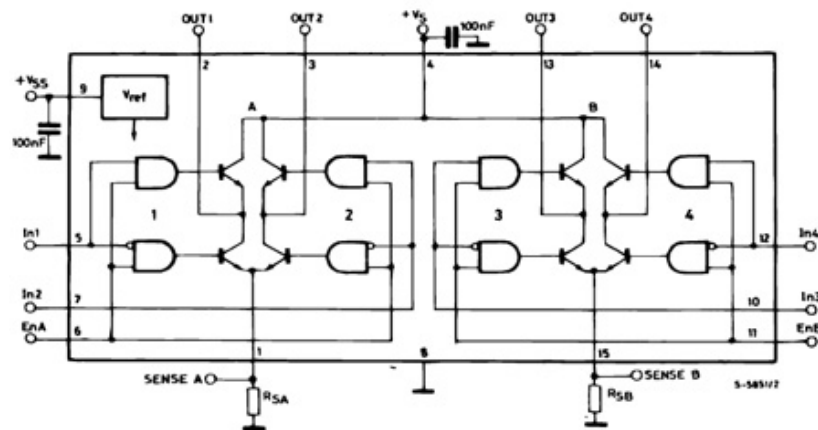


Figura 1 - Esquemático L298N [5]

2.3 DENAVIT-HARTENBERG

Existem alguns métodos para o cálculo da cinemática do robô, sendo o utilizado neste trabalho o de Denavit-Hartenberg descrito em [6]. Este método é baseado numa convenção para transformar sistemas de coordenadas. Um sistema de coordenada é transformado no seguinte através de uma matriz de transformação. Essa matriz é calculada a partir de quatro parâmetros, os parâmetros de Denavit-Hartenberg ou parâmetros DH. Os parâmetros DH são:

- d: distância através do eixo z antigo até a normal comum.
- r: distância entre os eixos z antigo e novo através da normal comum.
- θ : ângulo em torno do eixo z antigo entre o x antigo e a normal comum.
- α : ângulo em torno da normal comum entre os eixos z antigo e novo.

$$DH = \begin{bmatrix} \cos \theta & -\sin \theta * \cos \alpha & \sin \theta * \sin \alpha & r * \cos \theta \\ \sin \theta & \cos \theta * \cos \alpha & -\cos \theta * \sin \alpha & r * \sin \theta \\ 0 & \sin \alpha & \cos \alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\quad 1 \quad)$$

A Figura 2 mostra um exemplo do método sendo aplicado em juntas arbitrárias. Pode-se ver na imagem que os parâmetros são obtidos das relações entre os eixos z das duas juntas, sendo necessário apenas esses eixos, e o x da junta original, assim, é possível com a equação destes eixos automatizar o cálculo da matriz DH.

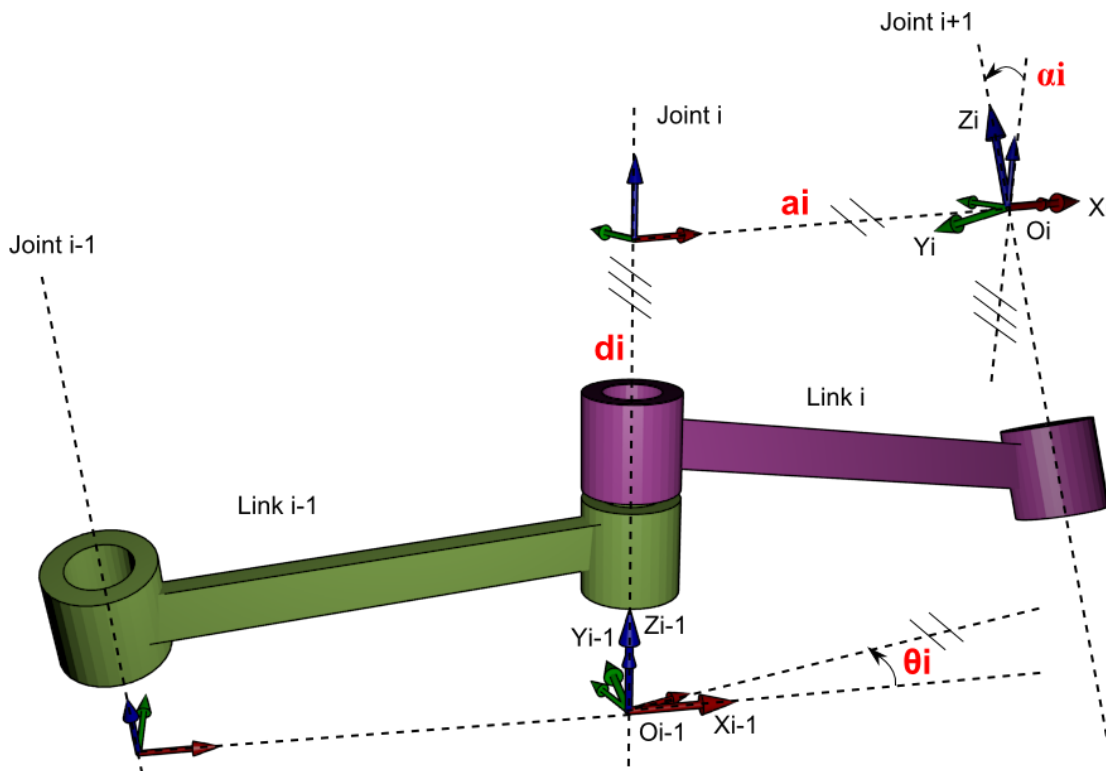


Figura 2 - exemplo de cálculo dos parâmetros DH [7].

3 MEDIÇÃO GEOMÉTRICA

Para a análise de um robô e de suas características estruturais é necessário obter os parâmetros referentes aos diversos componentes das estruturas mecânicas do robô. É de interesse saber se as peças foram produzidas conforme suas especificações e também como os desvios nas dimensões de cada peça podem influenciar nas características finais do robô. Outras informações de interesse para a análise estrutural do robô são os desvios de projeto devido a montagem.

Esse estudo engloba verificar as posições relativas entre as peças bem como as orientações. Erros de posicionamento na montagem podem levar a erros em parâmetros da cinemática do robô, enquanto que erros de orientação podem afetar toda a modelagem cinemática do robô. Quanto aos erros de orientação dois erros que exigem um estudo mais aprofundado são os erros de ortogonalidade e paralelismo, pois pequenos desvios de orientação em peças que deveriam ser ortogonais ou paralelas umas às outras podem gerar um modelo cinemático muito diferente.

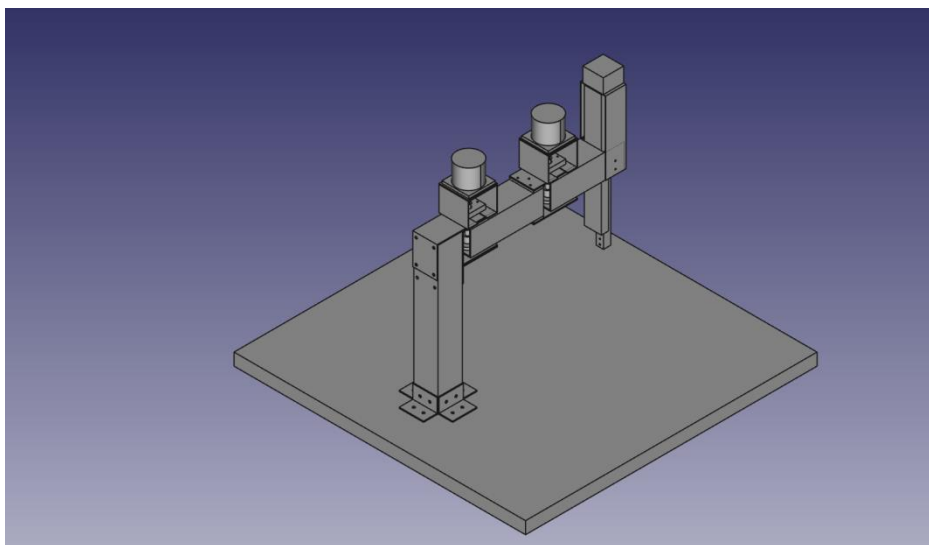


Figura 3 – Robô SCARA.

O robô a ser analisado nesse experimento é um robô do tipo SCARA, conforme o desenho da Figura 3. Este tipo de manipulador é projetado para ter movimento horizontal independente do movimento vertical. O movimento desta topologia é realizado por duas

juntas de rotação na horizontal e uma de translação na vertical. Um robô SCARA ideal tem os três eixos ortogonais ao eixo da base e assim paralelos entre si. Erros na produção e montagem podem afetar as distâncias entre as juntas, o paralelismo das juntas e a ortogonalidade entre as juntas e o plano da base do robô.

3.1 BRAÇO DE MEDIÇÃO

Existem diversos métodos para aferir as características estruturais do robô, as medições podem variar desde técnicas mais simples com instrumentos como paquímetros, esquadros, goniômetros, entre outros a métodos mais avançados empregando máquinas de medição de coordenadas. Para este trabalho foi utilizado o braço de medição ROMER Arm 100 (Figura 4) do Laboratório de Metrologia do Departamento de Engenharia Mecânica da UnB. A Tabela 2 apresenta as especificações deste braço de medição. Principais motivos para a escolha deste equipamento foram: a necessidade de medidas de precisão para se montar um modelo preciso do robô; a disponibilidade do equipamento no laboratório de metrologia; e a utilização do braço em um projeto parecido sobre outro robô didático.

Modelo	ROMER Arm 100
Eixos	6
Capacidade de medição radial	1250mm
Incerteza erro de apalpação	$\pm 16\mu\text{m}$
Incerteza medição volumétrica	$\pm 60\mu\text{m}$
Apalpador utilizado	Diâmetro 6mm

Tabela 2 - Características do Braço de Medição.

O braço de medição como o nome sugere é um braço robótico utilizado para se fazer medidas no espaço tridimensional. O braço utilizado é um braço articulado de seis eixos tendo três juntas de revolução para posicionamento e três juntas de revolução para orientação. Ao se posicionar o apalpador (ferramenta no efetuador terminal do braço) em um ponto de interesse pode-se saber a posição do ponto e a orientação que o braço toca o ponto através das variáveis de juntas do braço. Além da cinemática inversa que calcula a postura (posição e orientação) do efetuador terminal através das variáveis de juntas o braço também é calibrado para se ter maior precisão.



Figura 4 - Braço de medição.

3.2 COLETA DE DADOS

Quando o robô foi medido, este estava desenergizado, assim, as juntas não estavam travadas em uma posição específica. Para realizar as medições sem que o manipulador se movesse e desta forma introduzisse erro nos dados foi necessário trava-lo. Para travar as juntas o robô foi posicionado de forma a todas as juntas estarem no fim de curso e durante o processo de medição cada junta estava pressionada (manualmente) contra seu fim de curso para garantir que não haveria movimento.

A coleta de dados utilizando o braço de medição segue um funcionamento bem simples. Primeiramente se determina a entidade geométrica desejada (ponto, reta, plano, círculo). Em seguida obtêm-se os pontos pertencentes a essa entidade por meio da apalpação de cada ponto com o efetuador terminal do braço robótico. Por fim através dos pontos obtidos o software do braço calcula em relação ao seu sistema de coordenadas as características da entidade escolhida. Para entidades sobre determinadas como, por exemplo, um plano feito a partir de quatro pontos o plano calculado será o de erro quadrático dos pontos em relação a entidade (o plano) mínimo.

Antes de se coletarem os dados é interessante levantar quais dados são de especial interesse e para que são interessantes. É necessário obter o plano da base do robô uma vez que, através dele, pode-se fazer um sistema de coordenadas independente do sistema de coordenadas do braço de medição, o que possibilita o tratamento de dados mesmo que o robô esteja sendo medido com uma certa inclinação devido à mesa, por exemplo. Outras três entidades essenciais para modelar o robô são os eixos das juntas, definindo-se as retas

associadas a estes eixos, pode-se determinar distância entre eixos, paralelismo entre eixos, ortogonalidade entre eixo e os parâmetros necessários à modelagem do robô. Outros parâmetros também são de interesse para um estudo das fontes de erros mecânicos do robô. São estes a ortogonalidade das faces da haste de sustentação em relação à base do robô e o paralelismo entre as bases dos links entre juntas.

3.2.1 BASE

Para obter um plano a partir da base do robô e então usá-lo para montar um sistema de coordenadas de referência, foram obtidos quatro pontos por apalpação dos quais se calculou o plano da base, apesar de ser possível obter o plano com três pontos, se utilizaram quatro pois assim o resultado corrige melhor a não planicidade da base do robô. A Figura 5 ilustra o processo, onde p1, p2, p3 e p4 representam os pontos obtidos e pm é um ponto central onde é aplicado um vetor de direção, com pm e o vetor de direção é possível determinar o plano obtido. Como o plano foi obtido a partir de quatro pontos é possível que nenhum deles esteja efetivamente no plano já que o plano é obtido de forma a minimizar o erro quadrático médio dos pontos ao plano. O software do braço de medição é capaz de exportar em forma de planilha os dados sobre as estruturas medidas. Na Tabela 3 é possível ver uma parte da planilha onde se descreve o plano da base. Na coluna “name” está o nome dado à estrutura através do software, na coluna “type” o tipo de estrutura (“plan” para plano, “point” para ponto...), nas colunas x, y e z são dadas as coordenadas da estrutura em relação ao sistema de coordenadas do braço de medição e nas colunas a, b e c a orientação. É interessante notar que as orientações dos pontos se referem a informação de orientação do braço de medição e são, portanto, pouco informativas para a tarefa aqui proposta. Na Tabela 3 o ponto pm e o vetor de orientação são descritos na linha p1 com o tipo “plan”, pois eles definem o plano, as linhas seguintes informam os pontos usados para se calcular este plano.

Name	Type	X (mm)	Y (mm)	Z (mm)	A (rads)	B (rads)	C (rads)
P 1	Plan	-724,072	205,3281	-172,916	0,017266	-0,00623	0,999832
	Point	-515,86	-138,616	-175,696	0,250056	-0,266	0,930976
	Point	-386,066	401,518	-174,53	0,317723	-0,45978	0,82925
	Point	-1060,85	23,77217	-165,145	0,126315	-0,54614	0,828117
	Point	-933,305	534,5633	-164,296	-0,13156	-0,36219	0,922773

Tabela 3 - Dados plano base.

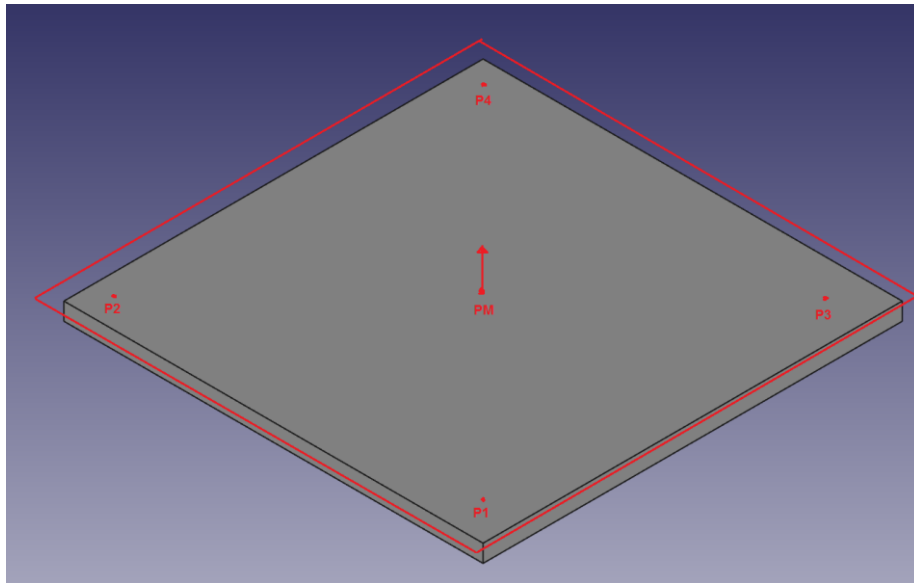


Figura 5 - Ilustração do plano de base obtido através do braço de medição.

3.2.2 HASTE DE SUSTENTAÇÃO

A haste de sustentação, como nome já diz é responsável por sustentar as demais partes do robô (excluindo a base), a princípio a haste deveria ser ortogonal ao plano da base, pois assim a estrutura a qual ela sustenta moveria num plano paralelo ao da base. As medições dessa parte do robô têm um caráter investigativo, uma vez que fornecerão mais informações de possíveis fontes de erro do que informações de modelagem. Também será usado o vetor diretor ortogonal à face direita da haste, para definir a partir da projeção desse vetor no plano da base a direção x do sistema de coordenadas da base. Assim como no ponto anterior através do plano da base já se tinha a direção z , agora com a direção x também definida através de um produto vetorial se tem também a direção y , assim, o sistema de coordenadas é totalmente definido. A obtenção dos planos de cada face da haste de sustentação foi feita de forma semelhante à que foi feita na base, para cada face se apalparam quatro pontos e extraiu-se um plano conforme a figura 4.

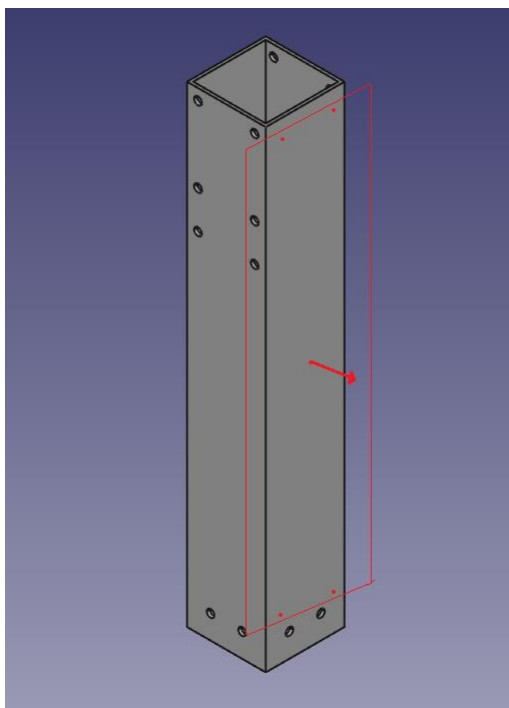


Figura 6 - Ilustração de um dos planos da haste de sustentação obtido através do braço de medição.

3.2.3 BASE DOS ELOS

Os últimos planos obtidos são os da base dos elos entre juntas e esta medida tem caráter investigativo visto que não contribui de forma direta para modelagem do robô. A Figura 7 nomeia cada uma das faces da haste de sustentação bem como enumera os links, o que vai facilitar a comunicação numa análise posterior. Os planos de cada link foram obtidos da mesma forma que os outros planos já obtidos nesse trabalho.

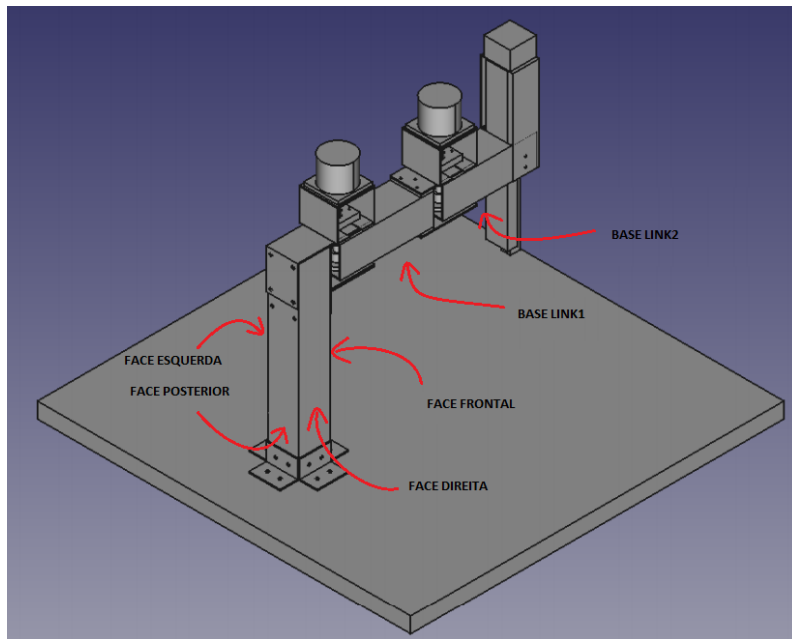


Figura 7 - Nomenclatura de algumas faces de estruturas do robô.

3.2.4 EIXO JUNTAS

As últimas estruturas obtidas, mas não menos importantes, são os eixos de cada uma das juntas, são essas as estruturas que possibilitarão a modelagem da cinemática do robô, pois as juntas são as responsáveis pelo movimento do mesmo. Como se pode ver na Figura 8, a região exposta do eixo do robô é muito pequena, além disso o acesso a essa região não é fácil sendo necessário posicionar o robô de forma que ele exponha a junta e também fique estável para não se mover entre a obtenção de um ponto e outro o que significa que esses pontos não foram obtidos na posição da Figura 7. Todos os fatores apresentados fazem com que essa parte da medição seja a mais complicada, no entanto foi realizada com sucesso. Para cada junta (principalmente a primeira e a segunda, as mais próximas da haste de sustentação) obtiveram-se os pontos conforme ilustrado na Figura 8, isto é, a partir de um círculo na região inferior e outro na região superior da junta, para cada círculo se precisou apalpar três pontos, também representados na Figura 8. A partir dos círculos extraiu-se uma reta passando pelo centro dos dois círculos a reta na imagem é representada como um vetor diretor aplicado a um ponto intermediário entre os dois centros. A reta foi assim representada pois é descrita assim no arquivo de planilha fornecido pelo software. O eixo de cada junta foi definido como sua respectiva reta obtida através desse método. Para a última junta (mais afastada da haste de sustentação) o método mediu nas extremidades do parafuso sem fim, tendo uma distância maior entre o círculo superior e o inferior.

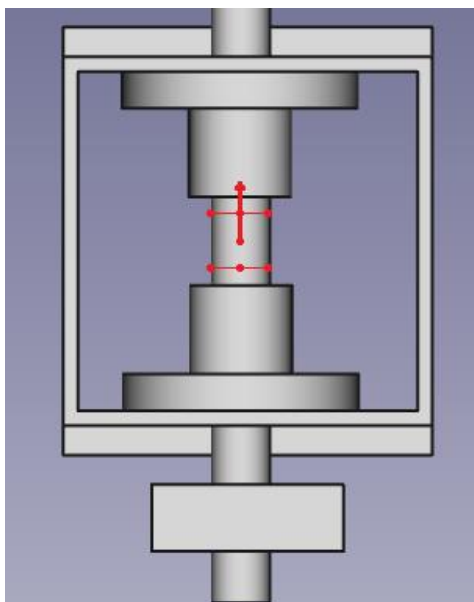


Figura 8 - Ilustração do eixo de uma junta obtido através do braço de medição.

3.3 ANÁLISE DE DADOS

Depois de obtidos os dados de cada estrutura a próxima etapa é a análise dos dados para obter informações de fato relevantes sobre o robô analisado. Para tal se utilizou do MatLab para, a partir dos devidos cálculos matemáticos, se obterem as relações entre as estruturas. O software do braço de medição é capaz de calcular algumas relações, mas se preferiu calcular tudo no MatLab para se ter maior precisão e também para se ter controle exato sobre o método de cálculo das relações. Tanto uma reta quanto um plano podem ser definidos com um ponto e um vetor diretor. Como o arquivo de planilha fornecem essas informações pode-se trabalhar apenas com elas, ignorando os pontos dos quais se extraiu a estrutura.

3.3.1 RELAÇÃO ENTRE PLANOS

As relações entre os planos são descritas na Tabela 4. Nela podem-se observar o valor medido, o valor esperado e o erro. O interesse para análise são as informações de ângulo entre os planos e, por isso, elas foram as informações medidas. É desejado saber se os planos que deveriam ser paralelos ($0^\circ/180^\circ$) entre si o são e se os planos que deveriam ser ortogonais (90°) o são. Para definir os ângulos entre os planos, calculou-se o ângulo entre os seus vetores de direção. Na Tabela 4, H.S. são os planos da haste de sustentação.

PLANO 1	PLANO 2	ÂNGULO MEDIDO	ÂNGULO DESEJADO	ERRO
Base	Frontal (H.S.)	89.5254°	90°	0.4746°
Base	Direito (H.S.)	89.4146°	90°	0.5854°
Base	Esquerdo (H.S.)	90.6156°	90°	-0.6156°
Base	Posterior (H.S.)	90.4379°	90°	-0.4379°
Base	Base link 1	179.5592°	180°	0.4408°
Base	Base link 2	178.1087°	180°	1.8913°
Frontal (H.S.)	Posterior (H.S.)	179.9364°	180°	0.0636°
Direito (H.S.)	Esquerdo (H.S.)	179.8119°	180°	0.1881°
Frontal (H.S.)	Base link 1	90.9154°	90°	-0.9154°
Frontal (H.S.)	Base link 2	91.3026°	90°	-1.3026°
Base link 1	Base link 2	1.7392°	0°	-1.7392°

Tabela 4 - Relações entre planos.

3.3.2 RELAÇÃO ENTRE PLANOS E RETAS

Na Tabela 5, podem-se ver os resultados apresentados de forma semelhante aos do item anterior. As relações de interesse entre retas e planos são primeiramente os ângulos entre os eixos de cada junta e o plano da base pois o modelo do sistema será formado a partir dessas relações e das relações entre juntas.

PLANO	RETA	ÂNGULO MEDIDO	ÂNGULO DESEJADO	ERRO
Base	Junta 1	0.5825°	0°	-0.5825°
Base	Junta 2	0.5131°	0°	-0.5131°
Base	Junta 3	0.9012°	0°	-0.5649°
Base link 1	Junta 1	179.2303°	180°	0.7697°
Base link 1	Junta 2	179.7979°	180°	0.2021°
Base link 2	Junta 2	178.4622°	180°	1.5378°
Base link 2	Junta 3	177.3753°	180°	2.6247°

Tabela 5 - Relações entre retas e planos.

O esperado era que todos os eixos fossem ortogonais à base ou seja o vetor direção das retas deveria ser paralelo ($0^\circ/180^\circ$) ao vetor direção do plano, uma vez que o vetor de direção do plano é ortogonal ao plano. Também é de interesse para uma análise dos desvios de montagem saber a relação de ângulo entre as juntas e os links entre elas que também deveriam ser ortogonais.

3.3.3 RELAÇÃO ENTRE RETAS

As retas (associadas às juntas) são o que vão definir a cinemática do manipulador, assim, é interessante saber as relações entre elas. A princípio, os eixos das juntas deveriam ser paralelos (0°) entre si. A Tabela 6 como nos itens anteriores mostra os ângulos entre as retas, A Tabela 7 mostra a distância entre as retas de duas formas: a menor distância entre as retas (que se dá fora da estrutura do robô) e a distância entre as retas num plano paralelo a base, que corta a junta 1 no meio do segmento de reta medido pelo braço de medição. A distância da Junta 1 até a Junta 3 não é a soma das distâncias entre as Juntas 1 e 2 e entre as Juntas 2 e 3 por que o robô não foi medido com os ângulos de juntas zerados.

RETA 1	RETA 2	ÂNGULO MEDIDO	ÂNGULO DESEJADO	ERRO
Junta 1	Junta 2	0.6592°	0°	-0.6592°
Junta 1	Junta 3	1.1857°	0°	-1.1857°
Junta 2	Junta 3	1.4140°	0°	-1.4140°

Tabela 6 - Ângulo entre retas.

RETA 1	RETA 2	MENOR DISTÂNCIA	DISTÂNCIA NO PLANO
Junta 1	Junta 2	108.1638	169.6193
Junta 1	Junta 3	50.1118	170.9765
Junta 2	Junta 3	48.4282	114.1769

Tabela 7 - Distância entre retas (mm).

4 MODELOS CINEMÁTICOS

Por meio da análise das relações entre as estruturas medidas, podem-se identificar vários desvios de projeto, o que se é esperado para um robô de fabricação não especializada. É necessário agora um estudo de como esses desvios afetam a cinemática do robô, os erros que eles introduzem no sistema e como corrigir esses erros.

O modelo cinemático é necessário para o entendimento do funcionamento do robô quanto a seu posicionamento. Este modelo relaciona as variáveis de junta com as variáveis de postura (posição e orientação) do efetuador terminal do robô, isto é, com a modelagem cinemática é possível saber a postura do efetuador terminal com base na posição de cada junta (cinemática direta), ou a partir da postura do efetuador terminal saber as variáveis de junta (cinemática inversa).

4.1 CINEMÁTICA DIRETA

A cinemática direta, como já dito, é uma função que recebe as variáveis de juntas e retorna variáveis de postura, existem dois principais métodos para se fazer o cálculo da cinemática direta. O primeiro é o método geométrico, a partir de desenhos esquemáticos se realizam cálculos de relações geométricas a fim de obter a cinemática direta do manipulador outro método bastante utilizado é o Denavit-Hartenberg.

Embora o método geométrico possa ser mais intuitivo, o método de Denavit-Hartenberg é melhor para se fazer um cálculo automatizado por ter instruções mais claras e objetivas. Para esse trabalho, foi feita uma função que recebe informações dos dois sistemas de coordenadas origem, orientação de x e z em relação a um sistema global (braço de medição) do sistema antigo e origem e orientação do eixo z em relação ao mesmo sistema global do novo sistema e também o tipo de variável de junta e o seu valor. A função retorna a matriz DH (conforme a equação 1), a nova origem e o novo x em relação ao sistema global.

4.1.1 MODELO PROJETADO

O modelo projetado representa a cinemática do robô como ele foi projetado desconsiderando-se qualquer desvio de projeto na produção das peças ou na montagem. Como este modelo é teórico e não se obtiveram valores reais de medidas para eles, para aplicar a função descrita anteriormente no cálculo desse modelo foi necessário definir retas no MatLab que representem os eixos projetados do robô, ou seja, paralelas entre si e com as distâncias entre elas conforme o projetado. O modelo ficou como segue:

$$x = l_1 * \cos(\theta_1) + l_2 * \cos(\theta_1 + \theta_2). \quad (\quad 2 \quad)$$

$$y = l_1 * \sin(\theta_1) + l_2 * \sin(\theta_1 + \theta_2). \quad (\quad 3 \quad)$$

$$z = z_{sc}. \quad (\quad 4 \quad)$$

$$l_1 = 168, l_2 = 112. \quad (\quad 5 \quad)$$

Onde x, y e z são as coordenadas do efetuador final nos sistemas de coordenadas da base, θ_1 e θ_2 são os ângulos das juntas 1 e 2 respectivamente, z_{sc} é o deslocamento da junta e l_1 e l_2 são, respectivamente, as distâncias entre as juntas 1 e 2 e as juntas 2 e 3.

4.1.2 MODELO MELHORADO

Em termos de complexidade o modelo melhorado é o que segue ao modelo projetado pois ambos são bem similares, com a diferença de pequenas correções de valores constantes. O modelo melhorado foi o último modelo calculado pois ele visa, sem aumentar a complexidade, diminuir os erros entre o modelo projetado e o modelo real (próximo modelo a ser analisado). O objetivo desse modelo é ser simples e possuir uma cinemática inversa conhecida, mas sem introduzir erros significativos ao sistema. O modelo então ficou como segue:

$$x = l_1 * \cos(\theta_1) + l_2 * \cos(\theta_1 + \theta_2). \quad (\quad 6 \quad)$$

$$y = l_1 * \sin(\theta_1) + l_2 * \sin(\theta_1 + \theta_2). \quad (\quad 7 \quad)$$

$$z = z_{sc}. \quad (\quad 8 \quad)$$

$$l_1 = 169.6193, l_2 = 114.1769. \quad (\quad 9 \quad)$$

4.1.3 MODELO REAL

O modelo real como o nome já diz é o modelo que leva em consideração o robô real e todos os seus desvios de projeto, o modelo foi calculado a partir das obtidas com o braço de medição para cada eixo juntamente com o plano da base do robô. O modelo real é uma sequência de quatro matrizes DH, a primeira faz as rotações necessária para orientar o sistema da base de acordo com o sistema da junta 1, a segunda matriz faz a transformada do sistema da junta 1 para a junta 2, a terceira transforma do sistema da junta 2 para o da 3 e a última posiciona no ponto de interesse da junta 3 (considerado o efetuador final para esse modelo, já que para cada ferramenta a posição do efetuador final iria variar). As matrizes ficaram como segue:

$$dh0r = \begin{bmatrix} 0.1191 & -0.9928 & 0.0101 & 0 \\ 0.9929 & 0.1191 & -0.0012 & 0 \\ 0 & 0.0102 & 0.9999 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

$$dh1r = \begin{bmatrix} \cos(\theta_1 + 0.57) & -0.9999 * \sin(\theta_1 + 0.57) & -0.01151 * \sin(\theta_1 + 0.57) & 108.2 * \cos(\theta_1 + 0.57) \\ \sin(\theta_1 + 0.57) & 0.9999 * \cos(\theta_1 + 0.57) & 0.01151 * \cos(\theta_1 + 0.57) & 108.2 * \sin(\theta_1 + 0.57) \\ 0 & -0.01151 & 0.9999 & 11366.0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

$$dh2r = \begin{bmatrix} \cos(\theta_2 + 0.25) & -0.9997 * \sin(\theta_2 + 0.25) & -0.02468 * \sin(\theta_2 + 0.25) & 48.43 * \cos(\theta_2 + 0.25) \\ \sin(\theta_2 + 0.25) & 0.9997 * \cos(\theta_2 + 0.25) & 0.02468 * \cos(\theta_2 + 0.25) & 48.43 * \sin(\theta_2 + 0.25) \\ 0 & -0.02468 & 0.9997 & -7165.0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

$$dh2re = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -4190.1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

A partir dessas matrizes, pode-se fazer então o modelo cinemático direto real. A última junta tem função de translação então no modelo ela serve para transladar um ponto sobre o eixo dela podendo ser modelada então como um deslocamento a partir do sistema de base definido pelas matrizes, a cinemática fica então:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = dh0r * dh1r * dh2r * dh2re * \begin{bmatrix} 0 \\ 0 \\ z_{sc} \\ 1 \end{bmatrix} \quad (14)$$

4.2 COMPARAÇÕES

É importante se compararem os modelos para definir qual é mais viável de ser utilizado, a princípio a escolha óbvia seria o modelo real que apesar de mais complexo o cálculo da sua cinemática direta é factível com a capacidade computacional disponível para esse projeto, no entanto, não se conhece a cinemática inversa do modelo real e esta é necessária para o controle de posicionamento. A grande questão que diferencia os modelos é que idealmente o movimento no eixo z deveria ser independente do movimento no plano xy, o que não ocorre devido aos desvios de montagem, então separou-se a análise das diferenças dos modelos em duas partes: primeiro, as duas juntas responsáveis pelo movimento no plano e então, a junta responsável pelo movimento vertical.

4.2.1 JUNTAS DE PLANO

Para comparar os modelos quanto às juntas responsáveis pelo movimento no plano, obteve-se a saída de cada modelo para todas as posições de juntas possíveis e, a partir, desses dados calcularam-se os erros entre os modelos. Como a junta 1 varia de 0° a 180° e a junta 2 de -90° a 90° e a resolução dos motores é de 1.8°, tem-se 10201 combinações de posições para cada modelo. Devido ao grande volume de dados as comparações dos

modelos tomaram como base os erros mínimo, médio e máximo. Os dados das comparações foram representados de duas formas através de tabelas e de gráficos, a Tabela 8 e a Figura 9 mostram dados do erro entre os modelos projetado e real, a Tabela 9 e a Figura 10 mostram dados do erro entre os modelos projetado e melhorado, a Tabela 10 e a Figura 11 mostram dados do erro entre os modelos melhorado e real. Na Tabela 11 e no gráfico da Figura 12 são mostrados resumidamente as 3 comparações entre os modelos. Todos os dados estão em milímetros.

	Erro total	Erro vertical	Erro plano
Mínimo	2.7183	0.1934	2.6599
Médio	5.0085	3.4410	3.3914
Máximo	7.6324	6.7047	3.7936

Tabela 8 - Modelo projetado x modelo real (mm).

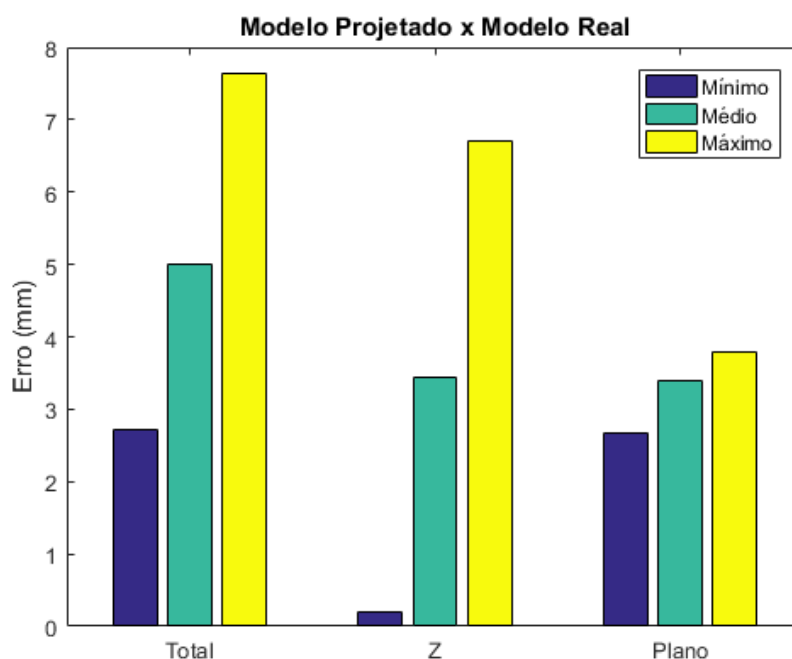


Figura 9 - Modelo projetado x modelo real (mm).

	Erro total	Erro vertical	Erro plano
Mínimo	4.3819	3.4410	2.7131
Médio	4.8571	3.4410	3.4197
Máximo	5.1236	3.4410	3.7962

Tabela 9 - Modelo projetado x modelo melhorado (mm).

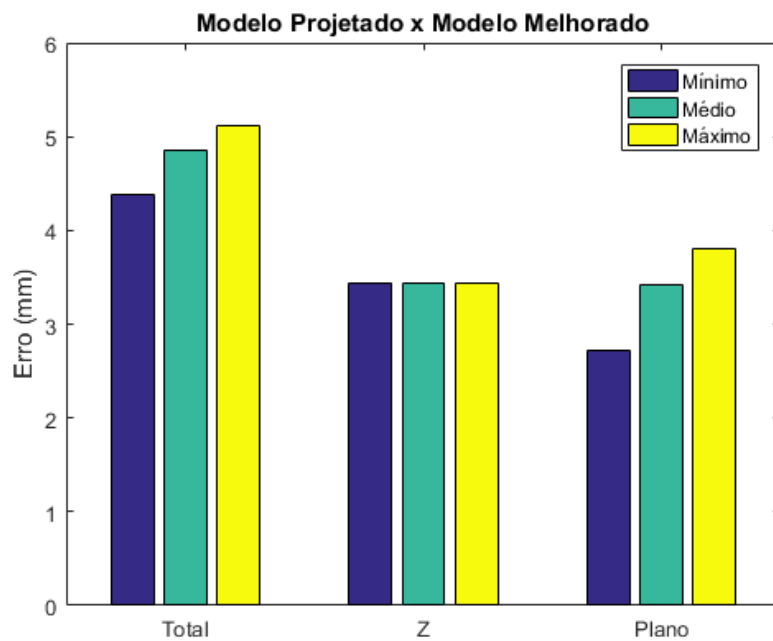


Figura 10 - Modelo projetado x modelo melhorado (mm).

	Erro total	Erro vertical	Erro plano
Mínimo	0.0331	0.0002	0.0006
Médio	1.5872	1.5859	0.0370
Máximo	3.2647	3.2637	0.0818

Tabela 10 - Modelo melhorado x modelo real (mm).

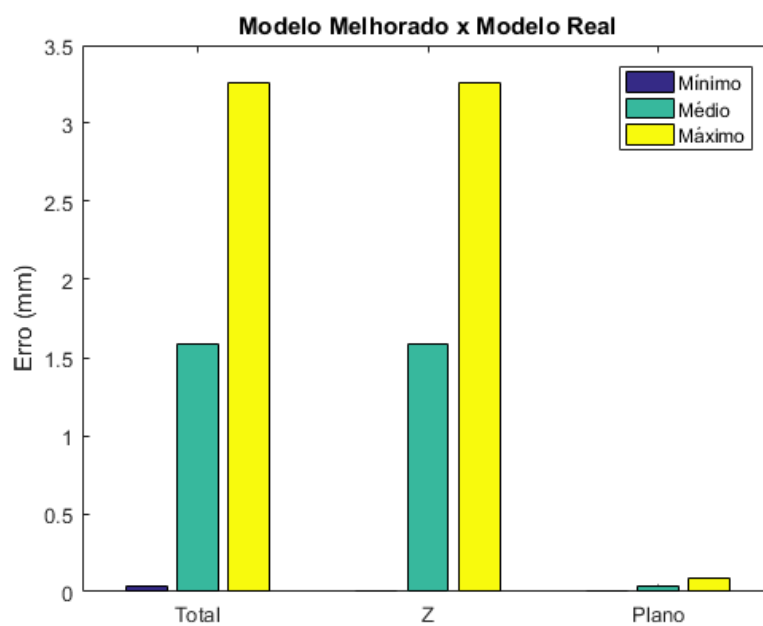


Figura 11 - Modelo melhorado x modelo real (mm).

	Projetado x Real	Projetado x Melhorado	Melhorado x Real
Mínimo	2.7183	4.3819	0.0331
Médio	5.0085	4.8571	1.5872
Máximo	7.6324	5.1236	3.2647

Tabela 11 - Comparação entre modelos (mm).

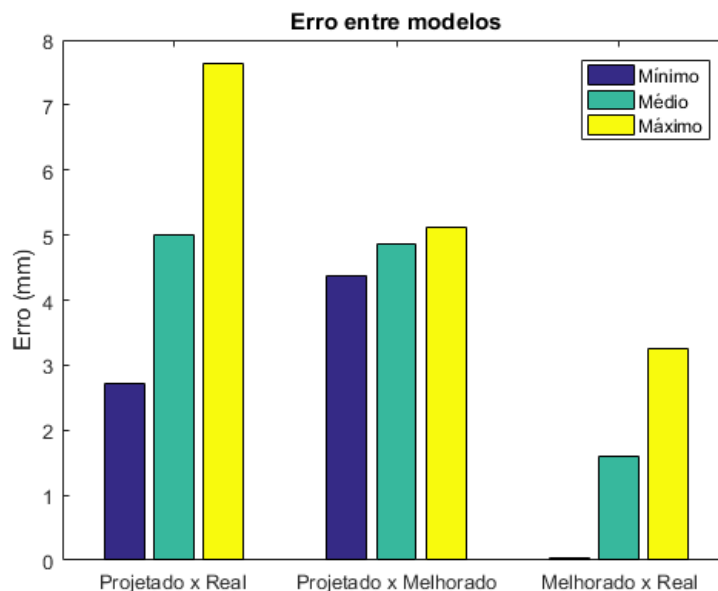


Figura 12 - Comparação entre modelos (mm).

4.2.2 ANÁLISE DE RESULTADOS DAS JUNTAS DE PLANO

Analisando-se os dados obtidos das comparações entre os modelos, fica claro duas coisas. Primeiramente é perceptível que o modelo melhorado consegue se aproximar significativamente melhor do modelo real se comparado com o modelo projetado. É também possível verificar que mesmo se trabalhando apenas com as juntas responsáveis pelo movimento no plano, a maior fonte de erro entre o modelo melhorado e o real é devido ao movimento no eixo z que no melhorado não existe, mas é verificado no modelo real. É interessante fazer um cálculo para verificar se o erro em z está dentro de um valor esperado. Sabe-se que uma barra inclinada pode ser decomposta em duas componentes, pode-se decompor o braço do robô em uma componente do plano e outra vertical, assim, tem-se que a componente vertical vai ser dada por:

$$l_z = l * \sin \theta. \quad (15)$$

Onde θ é o ângulo de inclinação do eixo de rotação do braço e l o tamanho de braço assumindo $l = 280\text{mm}$ (valor de projeto) e $\theta = 0.5^\circ$ (valor próximo às inclinações das juntas

no modelo real) e assumindo a inclinação constante no braço (simplificando o braço para apenas uma junta), tem-se que o erro em z deveria ser da ordem de 2.44mm o que foi observado.

Apesar de o erro no plano ser muito inferior ao erro vertical é necessário verificar se o erro é desprezível para o robô, ou seja, se o erro é desprezível em relação a resolução do robô. A resolução do robô varia conforme sua posição, mas seu menor valor é quando $\theta_2 = 180^\circ$, pois nessa posição o efetuador está mais próximo possível do centro de rotação de θ_1 e a resolução é dada pela distância entre dois pontos em um círculo cujo o raio é a diferença entre os tamanhos dos links, ou seja, 55.4424 mm e a distância angular é o valor do deslocamento angular mínimo, isto é, 1.8° . A resolução mínima do robô é:

$$resolução = 55.4424 * \sqrt{2 * (1 - \cos(\theta))} = 1.7417mm. \quad (16)$$

O erro máximo no plano é pelo menos vinte vezes menor que a menor resolução do robô que se dá em uma posição que o robô se quer atinge, assim, não se justifica a utilização do modelo real para o cálculo da cinemática no plano sendo, assim, o modelo melhorado é suficiente.

O erro médio na vertical é o problema a ser tratado, pois é um erro muito maior que o do plano e afeta uma junta de maior precisão, portanto, é um erro com maior impacto no resultado final da análise do robô. É desejável evitar a necessidade de se fazer uma cinemática inversa pelas complicações matemáticas envolvidas, assim, uma primeira estratégia seria assumir que o movimento na vertical não altera a posição no plano (os erros são devidos ao fator contrário o movimento no plano alterando a posição vertical). Com essa estratégia pode-se calcular normalmente a posição das juntas de plano e então corrigir na junta prismática o erro em z. O erro máximo dessa correção no plano é dado por:

$$\max(Erro_{plano}) = \max(Erro_z) * \sin(\theta) = 0.0322mm. \quad (17)$$

4.3 JUNTA VERTICAL

A Junta vertical que idealmente deveria deslocar o efetuador terminal apenas no eixo Z, mas devido aos desvios de montagem ela provoca também movimentação no plano XY é, então, importante calcular o valor desse deslocamento e ver se ele é relevante. O deslocamento pode ser calculado como:

$$\max(Deslocamento_z) * \sin(\theta) = 1.5728mm. \quad (\quad 18 \quad)$$

O impacto do deslocamento em Z no plano pode chegar a valores não desprezíveis, mas ainda assim é menor que a resolução mínima do robô.

5 MÉTODOS DE CONTROLE CINEMÁTICO

Depois da análise dos modelos do robô é necessário definir uma estratégia de controle. As variáveis que se conseguem controlar de forma direta, são as variáveis de juntas, pois seus valores são definidos conforme a posição dos motores de passo. Para controlar a posição do efetuador final é necessário resolver o problema da cinemática inversa, isto é, dado a posição desejada do efetuador terminal o método de controle deve calcular as variáveis de juntas que melhor se aproximam dessa posição. No entanto resolver analiticamente o problema da cinemática inversa não se mostrou viável devido à complexidade das equações do modelo real. Nesta seção é apresentado um método que busca resolver o problema da cinemática inversa de forma direta, isto é, um cálculo de cada variável de junta (uma solução aproximada) e um método de solução indireta onde é definido uma trajetória para o efetuador percorrer e, seguindo essa trajetória, o robô de forma indireta resolve o problema da cinemática inversa e por fim um método híbrido que tenta utilizar o melhor dos dois mundos.

5.1 MÉTODO DE PREVISÃO DE CORREÇÃO EM Z

Como visto no capítulo anterior o erro em Z é o principal fator de erro entre o modelo melhorado e o modelo real, assim, uma primeira estratégia para modelar a cinemática inversa é assumir que como deveria ser o movimento no plano é independente do movimento no eixo Z, assim para θ_1 e θ_2 a cinemática inversa seria como a do modelo melhorado que é conhecida e por fim corrigir o erro em Z ocasionado por essa movimentação. Com base na equação 15 pode se ver que o erro em Z (devido a θ_1 e θ_2) é dado por:

$$Erro_z = [0 \quad 0 \quad 1 \quad 0] * dh0r * dh1r * dh2r * dh2re * \begin{bmatrix} 0 \\ 0 \\ z_{sc} \\ 1 \end{bmatrix} \quad (19)$$

Fazendo $z_{sc} = 0$, pois o desejado é obter o erro em função de θ_1 e θ_2 é obtida a equação do erro em z:

$$\begin{aligned}
Erro_z = & 1.1 * \sin(\theta_1 - 0.57) - 1.3 * \cos(\theta_1 - 0.57) + 0.49 * \cos(\theta_2 + 0.25) \\
& * \sin(\theta_1 - 0.57) + 1.1 * \sin(\theta_2 + 0.25) * \sin(\theta_1 - 0.57) - 100.0 \\
& * \cos(\theta_2 + 0.25) * (0.01 * \cos(\theta_1 - 0.57) - 0.012) + 48.0 * \sin(\theta_2 \\
& + 0.25) * (0.01 * \cos(\theta_1 - 0.57) - 0.012) + 2.6.
\end{aligned} \quad (20)$$

Sabe-se que o eixo da junta prismática é inclinado e por isso para movimentar em Z é necessário um fator de correção para deslocar de forma que o deslocamento da junta prismática gere o deslocamento desejado em z ou seja:

$$Z_{sc} = \frac{(Z - Erro_z)}{\cos(\alpha_3)}. \quad (21)$$

Onde α_3 é o desvio do ângulo da junta 3. A partir dessas representações, a cinemática inversa por esse método pode ser calculada seguinte forma:

$$\theta_2 = \pm \cos^{-1} \left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2 * l_1 * l_2} \right). \quad (22)$$

$$\theta_1 = \tan^{-1} \left(\frac{y}{x} \right) - \tan^{-1} \left(\frac{l_2 * \sin(\theta_2)}{l_1 + l_2 * \cos(\theta_2)} \right). \quad (23)$$

$$Erro_z = [0 \quad 0 \quad 1 \quad 0] * dh0r * dh1r * dh2r * dh2re * \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (24)$$

$$Z_{sc} = \frac{(Z - Erro_z)}{\cos(\alpha_3)}. \quad (25)$$

$$l_1 = 169.6193, l_2 = 114.1769. \quad (26)$$

Para testar esse método proposto foi feita em MatLab uma malha de ponto de testes. A malha foi gerada através da fórmula da cinemática direta. Para as juntas 1 e 2, se variou o ângulo de junta de 0.9 em 0.9 graus a partir do valor mínimo até o máximo de cada junta como a resolução de junta do robô real é de 1.8 graus este método permite verificar o erro de posicionamento tanto para pontos alcançáveis, como para pontos inalcançáveis para o robô. Para a junta 3 se o variou a posição de 1 em 1 milímetro do seu valor mínimo ao máximo. A malha de teste contém 40.080.501 pontos.

O teste foi realizado de maneira a calcular a cinemática inversa de cada ponto e então calcular a cinemática direta a partir das variáveis de juntas resultantes da cinemática inversa

e então comparar a diferença do ponto de entrada para o teste e o ponto resultante da cinemática direta a partir das variáveis de juntas obtidas pela cinemática inversa. A Tabela 12 mostra os erros calculados sobre a malha sem quantizar as variáveis de junta. A Tabela 13 mostra os erros sobre a malha alcançável com quantização, ou seja, antes de se calcular novamente a cinemática direta, as variáveis de juntas foram quantizadas em valores múltiplos de 1.8°.

	Erro Total	Erro Plano	Erro Z	Erro Y	Erro X
Mínimo	0.0003	0.0003	0.0000	0.0000	0.0000
Médio	1.4089	1.4087	0.0199	0.9444	0.8604
Máximo	4.4977	4.4968	0.0926	3.6838	4.3210

Tabela 12- Erro malha intermediária sem quantização.

	Erro Total	Erro Plano	Erro Z	Erro Y	Erro X
Mínimo	0.5639	0.5638	0.0000	0.0000	0.0000
Médio	2.8208	2.8207	0.0200	1.8201	1.7653
Máximo	8.4673	8.4670	0.0928	8.4571	7.0635

Tabela 13- Erro malha intermediária com quantização.

Pela comparação das tabelas é fácil ver que a quantização causa grande impacto no erro do robô praticamente dobrando o erro. Apesar da precisão em Z temos este método insere erros não desprezíveis no sistema.

5.2 MÉTODO DE TRAJETÓRIA POR JACOBIANO INVERSO

Este método descrito em [8] é baseado no fato de que para pequenas variações as seguintes equações são verdadeiras:

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \approx \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \frac{\partial x}{\partial z_{sc}} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial y}{\partial z_{sc}} \\ \frac{\partial z}{\partial \theta_1} & \frac{\partial z}{\partial \theta_2} & \frac{\partial z}{\partial z_{sc}} \end{bmatrix} * \begin{bmatrix} \Delta \theta_1 \\ \Delta \theta_2 \\ \Delta z_{sc} \end{bmatrix}. \quad (27)$$

$$\begin{bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \Delta z_{sc} \end{bmatrix} \approx \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \frac{\partial x}{\partial z_{sc}} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial y}{\partial z_{sc}} \\ \frac{\partial z}{\partial \theta_1} & \frac{\partial z}{\partial \theta_2} & \frac{\partial z}{\partial z_{sc}} \end{bmatrix}^{-1} * \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}. \quad (28)$$

O método é melhor aplicado em geração de trajetória porque com uma solução direta da cinemática inversa envolveria grandes deslocamentos e assim o erro de aproximação seria grande, mas se a partir de uma trajetória desejada se discretizar o caminho até o ponto objetivo gerando uma sequência de pontos pertencentes a essa trajetória e com uma pequena variação de posição, assim, o algoritmo consegue funcionar melhor. Os termos do jacobiano são também aproximação, na verdade são uma divisão de diferenças como exemplo de um dos elementos da matriz temos:

$$\frac{\partial x}{\partial \theta_1} = \frac{(x(\theta_1 + 0.001, \theta_2, z_{sc}) - x(\theta_1 - 0.001, \theta_2, z_{sc}))}{0.002} \quad (29)$$

O apesar de ser um método aplicado à geração de trajetória, ele é aplicável também a uma resolução direta através de um método numérico. Como o método já usa a cinemática direto tanto para cálculo do jacobiano como para estimar a posição real é possível simular a execução do algoritmo sem efetivamente controlar o robô e então no fim da iterações utilizar as variáveis de juntas finais como o resultado da cinemática inversa, este método é interessante pois pode seguir a trajetória em simulação sem se preocupar com as limitações físicas do robô, por exemplo com a quantização das variáveis de juntas que só seriam levadas em contas depois de já obtido o resultado da cinemática inversa.

Por corrigir numericamente o erro é esperado que este método obtenha resultados, mas satisfatórios que o anterior para o cálculo da cinemática inversa do robô, no entanto, este método apresenta algumas desvantagens também. Por utilizar a inversa da jacobiana existe pontos onde a jacobiana não é invertível ou a inversa vai provocar deslocamentos muito altos tornando os erros de aproximação não desprezíveis, portanto caso o caminho até a posição final do robô através deste método passe por um desses pontos o método diverge. Com a inversa da jacobiana o resultado depende do ponto inicial de cálculo pois ele vai determinar o caminho. Por ser uma aproximação numérica e funcionar melhor com pequenos deslocamentos este método é mais custoso computacionalmente.

5.3 MÉTODO HÍBRIDO

Em virtude das vantagens e desvantagens de cada método de cálculo da cinemática inversa, é proposto um método híbrido onde se busca obter o melhor dos dois mundos, isto é, o caráter determinístico e a capacidade de fornecer resultados para toda a área de trabalho do método da correção em z aliados a maior precisão do método da inversa do jacobiano. O método híbrido consiste em calcular as variáveis de juntas pelo método da correção em z e então usar a o método da inversa do jacobiano para refinar o cálculo, inicialmente a melhor aproximação é salva como aquela obtida pelo método da correção em z, e então, a cada iteração caso se encontre uma melhor aproximação está se torna a nova melhor aproximação, assim, no pior caso o resultado desde método será igual ao da correção em z e no melhor caso será o quão bem o método da inversa do jacobiano conseguir aproximar. A Tabela 14 e a Tabela 15 mostram os resultados deste método aplicado à mesma malha de teste a qual se aplicou o método da correção em z. Sem a quantização esse método produziu resultado muito melhores que o da correção em z visto que o erro médio total é cerca de 485 vezes menor. Com a quantização a diferença entre os métodos não é tão grande, mas continua considerável com o erro máximo total caindo em 2mm e o erro total médio caindo pela metade. Além de convergir para toda a área de trabalho do robô o método híbrido possui ainda outra vantagem em relação ao método da inversa do jacobiano, por seu caminho começar em um ponto mais próximo em relação ao ponto final o algoritmo é executado em menos iterações. Assim, sendo mais rápido.

	Erro Total	Erro Plano	Erro Z	Erro Y	Erro X
Mínimo	0.0000	0.0000	0.0000	0.0000	0.0000
Médio	0.0029	0.0029	0.0000	0.0019	0.0017
Máximo	1.0922	1.0920	0.0202	0.9596	0.5212

Tabela 14 - Erro malha de teste sem quantização com método híbrido.

	Erro Total	Erro Plano	Erro Z	Erro Y	Erro X
Mínimo	0	0	0	0	0
Médio	2.1709	2.1708	0.0167	1.3836	1.3799
Máximo	6.1988	6.1987	0.0605	6.1420	6.1984

Tabela 15 – Erro malha de teste com quantização com método híbrido.

Para uma última comparação foi obtido os dados de erros para pontos a partir da quantização da malha de teste (pontos que seriam obtidos pelo cálculo real da cinemática

inversa), sem a quantização o erro seria zero, mas com a quantização podem-se observar erros conforme a Tabela 16, é interessante notar que a aparente melhora nos resultados não quer dizer que método híbrido supere até mesmo o cálculo real da cinemática inversa, mas sim que para esses pontos e a quantização aplicada ele obteve um desempenho melhor, o que mostra que o desempenho deste método já é muito próximo dos limites impostos pela quantização.

	Erro Total	Erro Plano	Erro Z	Erro Y	Erro X
Mínimo	0	0	0	0	0
Médio	2.4973	2.4969	0.0154	1.5896	1.5896
Máximo	6.3347	6.3347	0.0609	6.3346	6.3345

Tabela 16 - Erro malha de teste com quantização com cálculo real da cinemática inversa.

6 GERAÇÃO DE TRAJETÓRIA

Para algumas aplicações basta somente que o robô se mova de uma posição para a outra não importando o caminho percorrido, mas existem também aplicações onde percorrer dado caminho é tão importante ou mais que chegar a posição final. Para aplicações onde o caminho é importante são utilizados algoritmos de geração de trajetória.

6.1 INTERPOLAÇÃO

A interpolação [8] é um conceito importante para a geração de trajetória, pois serve de base para alguns dos algoritmos mais utilizados nessa área. A interpolação consiste em transformar uma trajetória em uma sequência de pontos a serem alcançados pelo manipulador. A quantidade de pontos na interpolação é um fator que pode afetar o quão bem o manipulador segue essa curva, além de afetar a velocidade com a qual o manipulador percorre a trajetória.

6.2 JACOBIANO INVERSO

Apesar de ter sido utilizado como um método indireto de se resolver o problema da cinemática inversa neste trabalho, a técnica de aproximação por jacobiano inverso é em essência uma técnica de geração de trajetória. O Jacobiano precisa de uma interpolação da trajetória pois seus resultados são melhores para pequenos deslocamentos assim se faz necessário uma sequência de pontos a ser seguida calculando a distância entre a posição atual do manipulador e o próximo ponto desejado da sequência. É importante listar algumas características deste método:

- Custo computacional baixo por iteração.
- Não precisa da cinemática inversa.
- Necessita de uma interpolação com muitos pontos para um desempenho satisfatório.
- Pode divergir.
- Não garante que se chegue ao final da trajetória.

6.3 CINEMÁTICA INVERSA

Outra estratégia que pode ser adotada para geração de trajetória é calcular a cinemática para cada um dos pontos da interpolação da trajetória desejada. Este método tem como características:

- Custo alto por iteração.
- Necessita de um modelo cinemático inverso.
- Funciona para interpolação com poucos pontos
- Não diverge
- A chegada ao final da trajetória é garantida, ainda que passe por uma região não alcançável

6.4 CRITÉRIOS DA ESCOLHA

O método escolhido para se trabalhar nesse projeto foi o da interpolação com cinemática inversa, esta escolha foi baseada no fato de que este método é mais robusto e estável que a outra opção. O método escolhido além não ter pontos de divergência na região alcançável do manipulador, para tarefas que seguir trajetória seja desejável, mas não um fator crítico a possibilidade de se ter uma trajetória passando pela região não alcançável pode ser interessante. Caso seguir trajetória seja crítico não é difícil negar trajetórias que passem pela região não alcançável do manipulador. O principal ponto contra ao método escolhido é o alto custo por iteração, mas além de ser balanceado pelo fato deste poder utilizar uma interpolação com menos pontos que a outra opção, o poder computacional como será dito no capítulo 8 deste projeto faz com que não seja um problema o custo computacional.

7 PROJETO E IMPLEMENTAÇÃO ELETRÔNICA

O projeto eletroeletrônico define todo o hardware necessário para se fazer a atuação e o controle do manipulador, isto é, os controladores, os drivers de potências, os motores, os sensores e todo componente necessário para estas funções.

No trabalho anterior o robô foi feito completamente funcional, mas para esse trabalho se fez necessário um novo projeto eletroeletrônico, pois o hardware original não estava disponível. No lugar de apenas implementar um hardware igual ao do projeto original escolheu-se desenvolver um novo projeto com possíveis melhorias.

7.1 MOTORES

Os motores são os atuadores do robô, são eles os responsáveis por fornecer energia mecânica para mover todas as juntas. Os motores também são responsáveis por fazer com que as juntas fiquem paradas em uma posição resistindo a forças externas.

Os motores de passo utilizados no projeto anteriores continuaram disponíveis, sendo assim o projeto continuou fazendo uso deles. Todos os motores são motores de passo de acionamento unipolar de seis fios e foram montados de forma a serem acionados como motores bipolares para aumentar assim o torque produzido pelo motor.

Como descrito no trabalho anterior [1] os motores não apresentaram torque o suficiente para o acionamento em meio-passo sem perdas de passo. Perdas de passos não são aceitáveis, pois o robô não possui sensores e a estratégia de controle faz uso do posicionamento preciso do motor de passo para um controle em malha aberta. De forma igual ao projeto anterior foi definida uma estratégia de controle em passo completo, pois desta forma os motores conseguem atuar no robô sem perda de passos.

7.2 DRIVERS DE POTÊNCIA

Os drivers de potência são os responsáveis para fornecer energia elétrica aos motores com uma potência adequada para que estes possam atuar corretamente. Os drivers tem também a função de desacoplar eletricamente a parte lógica do sistema (micro controladores sensores e etc.) da parte de potência (motores e demais atuadores).

7.2.1 L298N

No projeto anterior foram utilizados drivers baseados no circuito integrado L298N (Figura 13) da SGS-ThomsonMicroeletronics [5] que tem cada as duas pontes H. As pontes H permitem definir o sentido da corrente o que é necessário para um acionamento em modo bipolar dos motores (que é o desejado nesse trabalho).

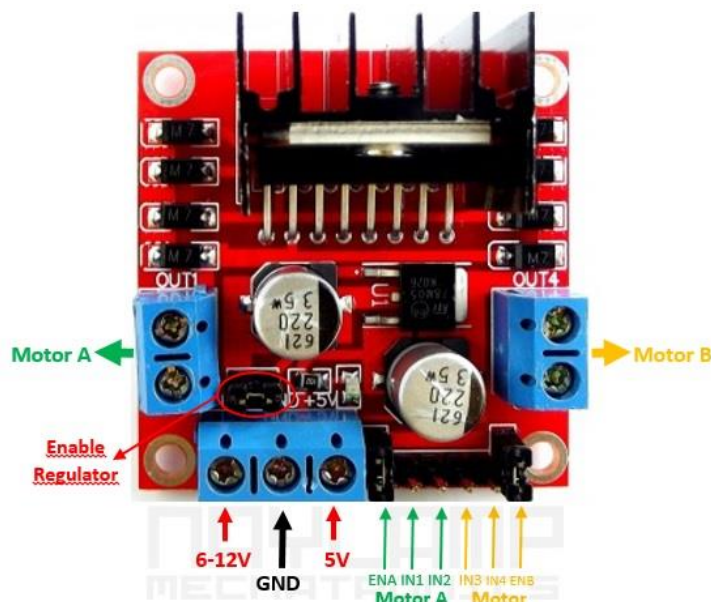


Figura 13 - Driver de potência baseado em L298N [9].

A placa baseada em L298N é uma placa genérica com duas pontes H que pode ser usada para diversos propósitos onde sejam necessários duas pontes H. Para controlar um motor de passo em modo bipolar com essa placa deve-se ligar uma bobina em cada saída de motor do driver (Motor A e Motor B na Figura 13), é necessário ainda que o controlador execute a lógica de acionamento da bobina chaveando uma bobina de cada vez. Para energizar uma bobina uma de suas entradas deve estar em nível alto lógico e outra em nível baixo lógico, assim, para chavear a bobina é necessário inverter o nível lógico de suas duas entradas. No método de acionamento escolhido nesse projeto sempre as duas bobinas estarão energizadas produzindo assim mais torque o chaveamento só definirá a direção da corrente na bobina. Um pwm nas entradas de habilitação permite controlar a tensão de saída quando se usa uma fonte mais potente.

7.3 TTL 7404

Para chavear as bobinas o par de entrada lógica de cada bobina sempre serão o negativo lógico um do outro, assim para economizar portas do controlador e não necessitar usar dois *arduin*os foi utilizado uma CI (Figura 14) com seis portas lógicas *not* que fornece o negativo lógico de suas entradas.

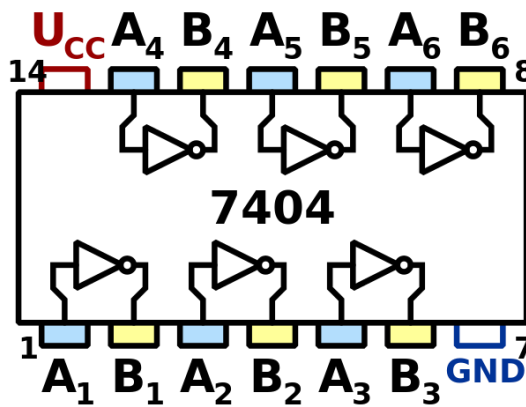


Figura 14 - Pinagem TTL 7404 [10].

7.4 CONTROLE

Para fazer o controle do robô no trabalho passado foi utilizado um computador ligado a dois micros controladores: *arduino* Uno Genuíno [11]. No trabalho anterior a função dos *arduin*os era realizar o correto chaveamento das bobinas, passando o comando para que o L298N acionasse os motores. Foi necessário dois *arduin*os para poder fazer o movimento síncrono das juntas rotacionais.

Nesse projeto com a porta *not* e simplificações no código *arduino* conseguiu se utilizar apenas um *arduino* para realizar o controle do robô. O *arduino* deve apenas receber um byte representando os motores a serem acionados e suas direções e passar o comando a cada motor, diferentemente do projeto anterior o *arduino* não vai receber nenhum comando mais complicado que dar um passo em cada motor. A Figura 15 mostra o fluxograma do projeto eletrônico.

7.5 FONTE

A energia para parte lógica é fornecida pelo computador conectado ao *arduino*, mas a para a parte de potência deve ser fornecida por uma fonte independente da lógica para não introduzir ruídos aos circuitos lógicos. Para o projeto se utilizou uma fonte de 20V e 3.5A (70W), os motores são de baixa tensão e alta corrente, então para fornecer a tensão e corrente correta cada placa L298 recebe um valor pwm que liga e desliga as pontes de forma a forma a diminuir a tensão e aumentar a corrente fornecida aos motores.

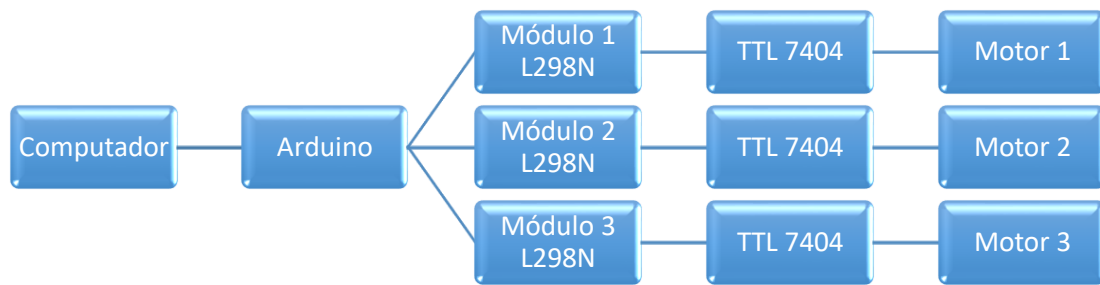


Figura 15 - Fluxografo parte eletrônica.

7.6 IMPLEMENTAÇÃO ELETRONICA

Para facilitar uma montagem mais compacta foi utilizado um protoshield conforme a Figura 16 este shield do arduino permite uma montagem compacta com uma protoboard em cima do arduino, além de ter a opção de retirar a protoboard e soldar elementos no próprio shield.

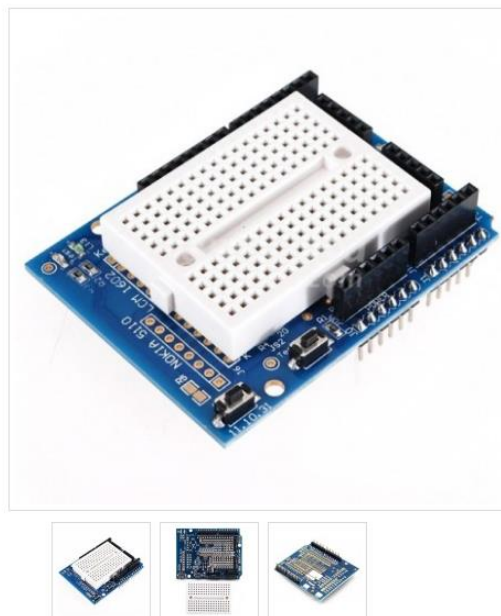


Figura 16 – Protoshield [12].

O sistema foi montado de forma a ter um TTL 7404 na protoboard. As saídas do arduino foram ligadas conforme a Tabela 17 e as placas conforme a Tabela 18. A Figura 17 mostra o sistema com caixa protetora aberta, a Figura 18 com a caixa fechada e a Figura 19 o arduino com a protoshield conectados no sistema. Em cada placa driver L298N os pinos “enable” das duas pontes H foram ligados juntos.

PIN 5	ENA/ENB L298N 3	PIN 10	ENA/ENB L298N 1
PIN 6	TTL 7404 A6	PIN 11	ENA/ENB L298N 2
PIN 7	TTL 7404 A3	PIN 12	TTL 7404 A2
PIN 8	TTL 7404 A5	PIN 13	TTL 7404 A1
PIN 9	TTL 7404 A4		

Tabela 17 - Conexões arduino.

L298N 1 ENA/ENB	PIN 10	L298N 2 IN3	TTL 7404 A1
L298N 1 IN1	TTL 7404 A5	L298N 2 IN4	TTL 7404 B1
L298N 1 IN2	TTL 7404 B5	L298N 3 ENA/ENB	PIN 11
L298N 1 IN3	TTL 7404 A4	L298N 3 IN1	TTL 7404 A6
L298N 1 IN4	TTL 7404 B4	L298N 3 IN2	TTL 7404 B6
L298N 3 ENA/ENB	PIN 5	L298N 3 IN3	TTL 7404 A3
L298N 2 IN1	TTL 7404 A2	L298N 3 IN4	TTL 7404 B3
L298N 2 IN2	TTL 7404 B2A		

Tabela 18 - Conexões lógicas L298N.

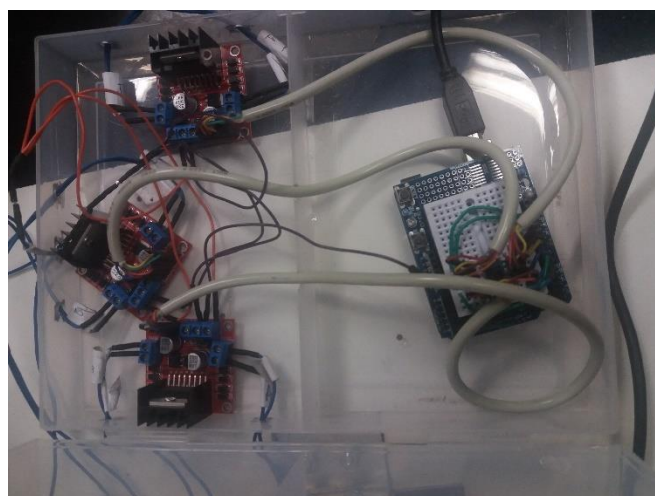


Figura 17 - Sistema eletrônico com caixa protetora aberta.

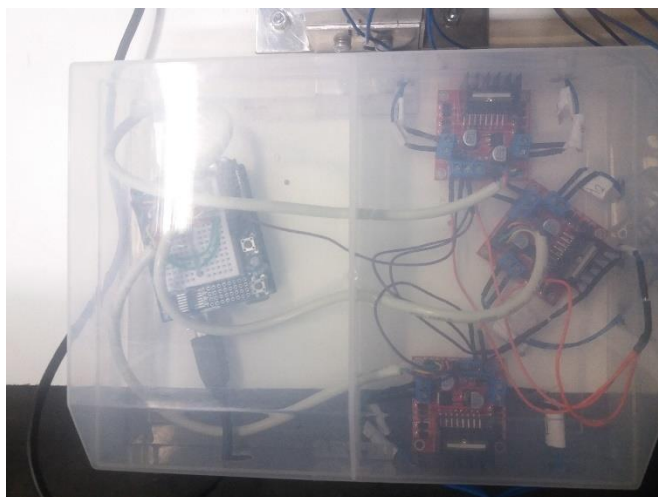


Figura 18 - Sistema eletrônico com caixa protetora fechada.

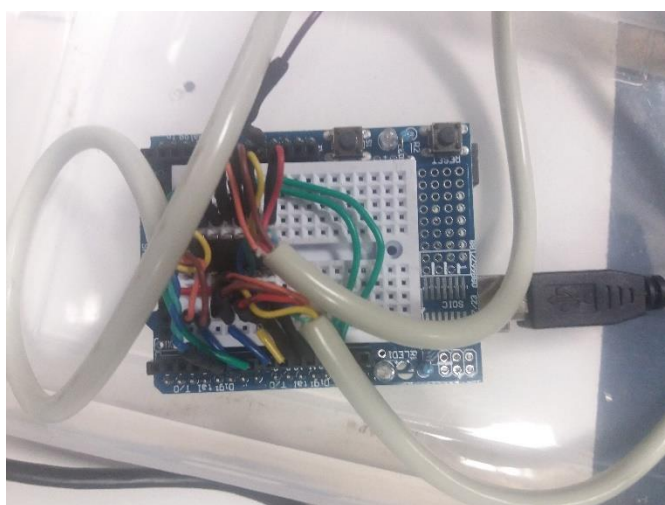


Figura 19 – Conjunto arduino e protoshield conectado ao sistema.

8 PROJETO E IMPLEMENTAÇÃO DE SOFTWARE

O projeto de software é o responsável por todo o trabalho computacional. Atuando com interface com o usuário, realização dos modelos e comunicação entre módulos de computação. O projeto software deve implementar o método de controle apresentado no capítulo 1 e a comunicação descrita no projeto eletrônico do capítulo 7 mais especificamente os requisitos de software são esses:

- Interface gráfica simples com o usuário.
- Entrada de dados on-line: modo passo e modo coordenadas.
- Interface para treinamento.
- Interface para configuração.
- Interface para importar e exportar trajetórias treinadas.
- Realizar algoritmos de controle.
- Gerar trajetórias.
- Comunicar PC e arduino.
- Acionar drivers de potência.

O projeto é dividido em dois módulos, dois programas [13] um para o arduino e outro para o PC.

8.1 MÓDULO ARDUINO

O programa do arduino deve receber um byte do PC interpretar esse Byte para saber os motores a serem chaveados e suas respectivas direções e chavear suas saídas de forma a fazer com que os motores deem um passo na direção desejada. O arduino precisa somente receber qual ou quais motores o módulo no PC deseja mover e enviar o comando para os drivers de potência, a Figura 20 ilustra o funcionamento do arduino.

O arduino recebe dados que são bytes na forma:

$$XXD_3S_3D_2S_2D_1S_1$$

Onde os X não representam nada, D_n é a direção do motor n e S_n indica se o motor n vai dar um passo. Se S_n é igual a 1 o motor deve dar um passo, ou seja, a saída incrementa como código gray (00 -> 01 -> 11 -> 10 -> 00...) caso D_n seja igual a 0 e decrementa (00 -> 10 -> 11 -> 01 -> 00....) caso D_n seja igual a 1. Os bits de saída representam os pinos do arduino responsáveis por cada bobina do motor. Para o motor 1 são os pinos 9 e 10, para o motor 2 os pinos 12 e 13 e para o motor 3 os pinos 6 e 7. Além disso tem um pwm sempre

no mesmo valor para cada motor o pino 10 para o motor 1 o 11 para o motor 2 e o 5 para o motor 3.



Figura 20 - Funcionamento arduino.

8.2 MÓDULO COMPUTADOR

O programa no computador fica com toda a carga de computação, ele que implementa a interface com usuário e o controle. A interface foi implementada através da biblioteca SDL [14] para C++, foi utilizada essa biblioteca por motivo de familiaridade e de fácil adaptação de códigos de outros trabalhos para esse. Para comunicação com o *arduino* foi utilizado um código disponível no site do próprio *arduino* [15]. A Figura 23. Mostra o funcionamento do software do computador.

8.2.1 INTERFACE GRÁFICA

A interface gráfica da SDL é orientada a objetos e neste projeto foi desenvolvida na forma de menu, o usuário pode navegar pelo menu através do teclado. Na tela de operação o usuário pode selecionar entre vários métodos de entrada de comandos, estes são, passos do motor, coordenadas, vetor de posição salva e entrada em código. A Figura 21 mostra a tela de operação no modo de coordenadas e a Figura 22 mostra a tela no modo de código. Além de operação existem também outros menus no programa para configurar o robô.

8.2.2 OPERAÇÃO DO ROBÔ

O programa permite vários modos de operação, pode se operar pelo menu e as caixas de texto onde se tem a opção de controle por passo dos motores, por coordenada e por um vetor de posição que deve ser criado no menu treinar, ou lido de um arquivo. Todo este modo tem entrada via caixas de texto e via teclado que é explicado no próprio menu do programa a função de cada tecla.

8.3 CONTROLE POR CODIGO

Foi definido uma estrutura de controle por código que pode funcionar de duas maneiras também implementadas no menu operar. O programa pode funcionar como um terminal onde o código é executado conforme digitado, ou ler um arquivo com o código, interpretar e executar.



Figura 21 - modo coordenadas



Figura 22 - modo código.

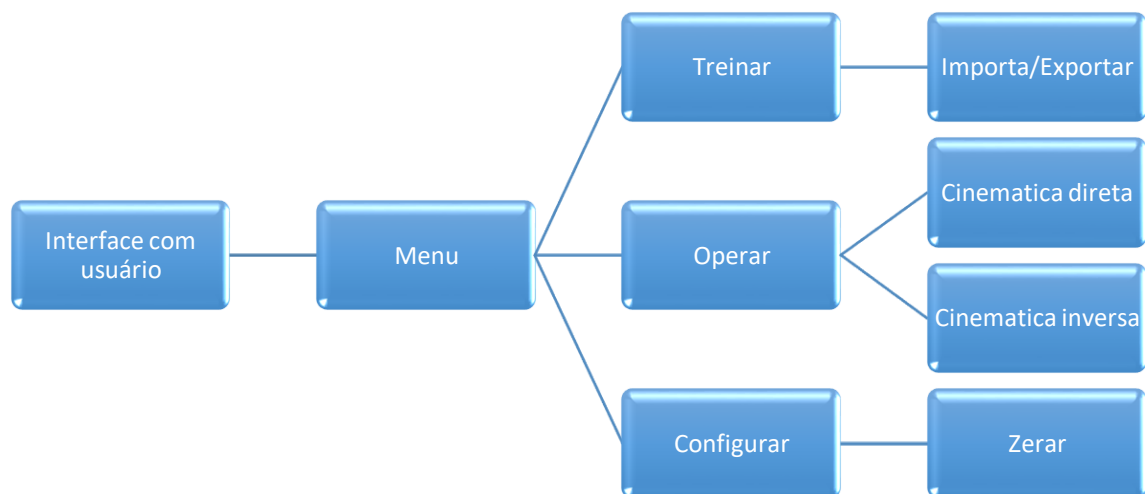


Figura 23 - Funcionamento PC.

O código lido pelo programa é bem simples cada linha tem um valor (caso haja separação por espaço o que vier depois do espaço é ignorado), o valor pode ser um comando ou um operando. Se o valor for um comando o comando anterior recebe os operandos e é executado. Enquanto o valor for um operando vai se armazenando os operandos em um vetor que é enviado como parâmetro para o comando atual quando um

novo comando for enviado. Caso algum operando não seja valido para o comando ele é ignorado. Como um caso especial o comando mov é executado (recebe os operandos) sempre que o vetor de operandos tiver tamanho 3. A Tabela 19 tem a lista de comandos.

modem	Configura para modo por motor (operandos serão entendidos como passos).
modec	Configura para o modo por coordenadas (operandos serão entendidos como valores de coordenadas).
Mov	Move para o ponto indicado
movr	Move para o ponto indicado em linha reta
movp	Move para o ponto indicado em trajetória polinomial
End	Executa o comando atual

Tabela 19 - Lista de comandos.

8.3.1 MODEM/MODEC

Os operandos podem ser entendidos como valor em passos de cada motor, ou como as coordenadas de um ponto, assim, modem e modc são usados para definir como o usuário irá trabalhar. A execução desses comandos é imediata e estabelece como comando atual o mov.

8.3.2 MOV

O mov é uma movimentação simples onde todos os motores são acionados ao mesmo tempo com as mesmas velocidades. Sempre que o comando atual for mov e houver três operandos no vetor ele é executado, caso o comando seja executado antes de ter três operandos ele completa os operandos restante como 0.

8.3.3 MOVR

O movr movimenta em reta entre os pontos escolhidos, quando executado recebe o vetor de operandos e completa com zeros até seu tamanho ser múltiplo de três. Cada três operandos define um ponto, o robô vai seguindo a ordem dos pontos movendo-se em linha reta para cada um. Após execução o comando atual é estabelecido como mov. O vídeo [16] mostra o comando em execução e a Figura 24 mostra a reta resultante.

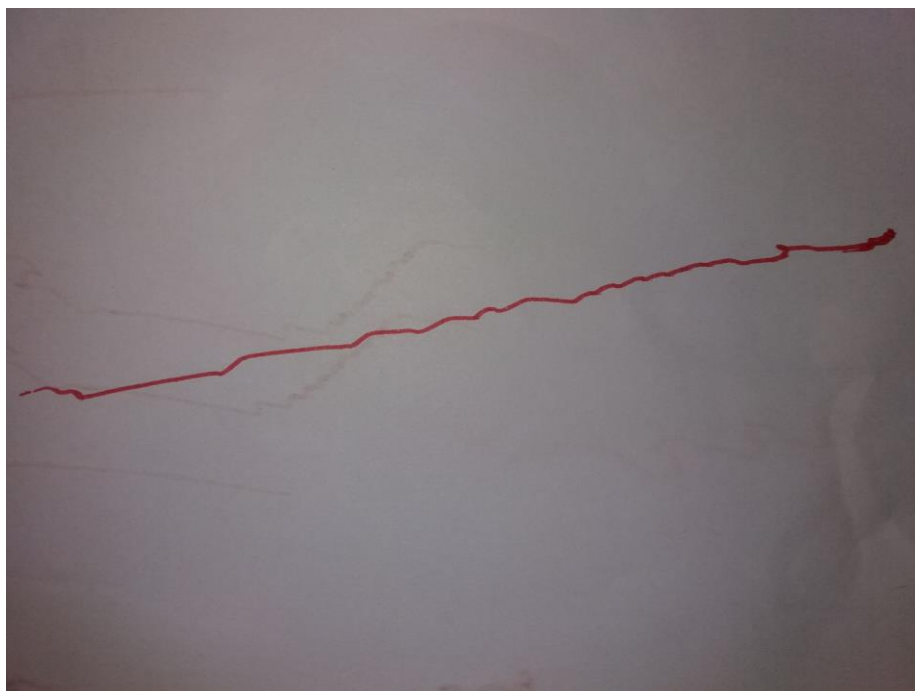


Figura 24 - Exemplo de reta com movr.

8.3.4 MOVp

O movp funciona de forma semelhante ao movr, porém, ao invés de se movimentar entre os pontos em linhas retas a trajetória de movimento é um polinômio de ordem igual ao número de pontos recebido que passa por todos os pontos terminando no último. O vídeo [17] mostra o comando em execução para dois pontos, assim, movendo em forma de parábola (polinômio de grau 2). A Figura 25 mostra o resultado onde nota-se que a parábola não foi realizada tão bem quanto a reta. A dificuldade em seguir a parábola é atribuída a dois fatores, primeiramente a parábola desejada é uma curva com concavidade contrária a concavidade do movimento natural do robô, além disso na seção onde o traço é reto atribui-se isso a parábola passar por uma região fora da área de trabalho do robô.

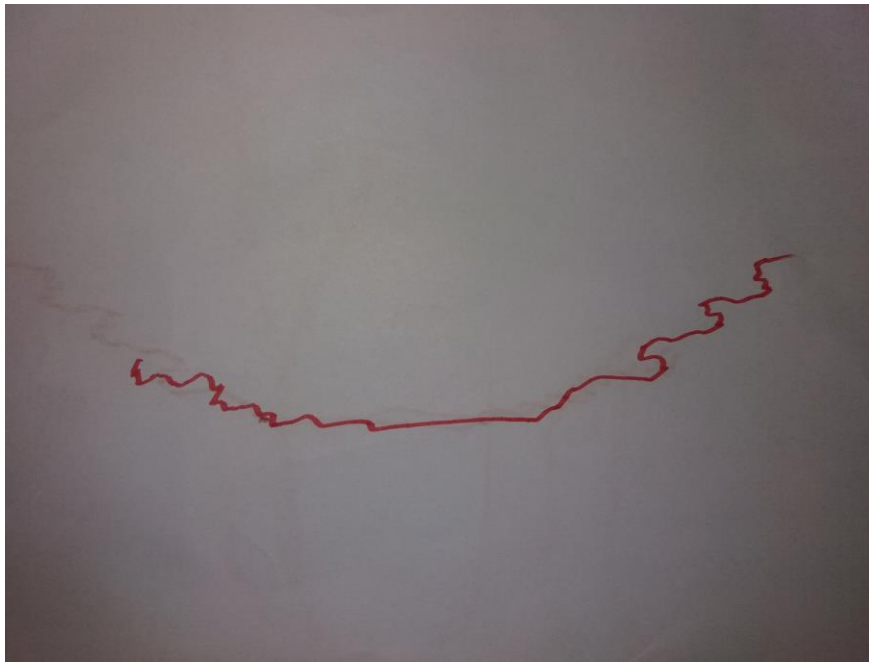


Figura 25 - Exemplo de parábola com comando movp.

8.4 CONTROLE

Para controle no modo passo é o valor informado pelo usuário é comparado com as restrições de valores para cada junta e então o software no PC envia comandos ao *arduino* até que esteja na posição desejada. Para o método por coordenadas é utilizado a cinemática inversa pelo método híbrido.

9 ANÁLISE DE DESEMPENHO

Para saber se um robô é adequado para determinada tarefa é importante se levantar suas características de desempenho, para isso se faz a análise de desempenho. Existem diversos critérios de análise para robôs manipuladores. Conforme a tarefa desejada determinado processo pode ser mais ou menos relevante. Existem dois conceitos [18] importantes para o estudo do desempenho do robô, pois estes são abordados por vários critérios de desempenho:

- Acuracidade: proximidade entre o valor obtido e o valor desejado de uma grandeza física
- Repetibilidade: desvio do valor obtido para um mesmo valor desejado.

A Figura 26 mostra graficamente o significado destes conceitos, que podem ser aplicados ao posicionamento do robô, a trajetória e outras características de interesse.

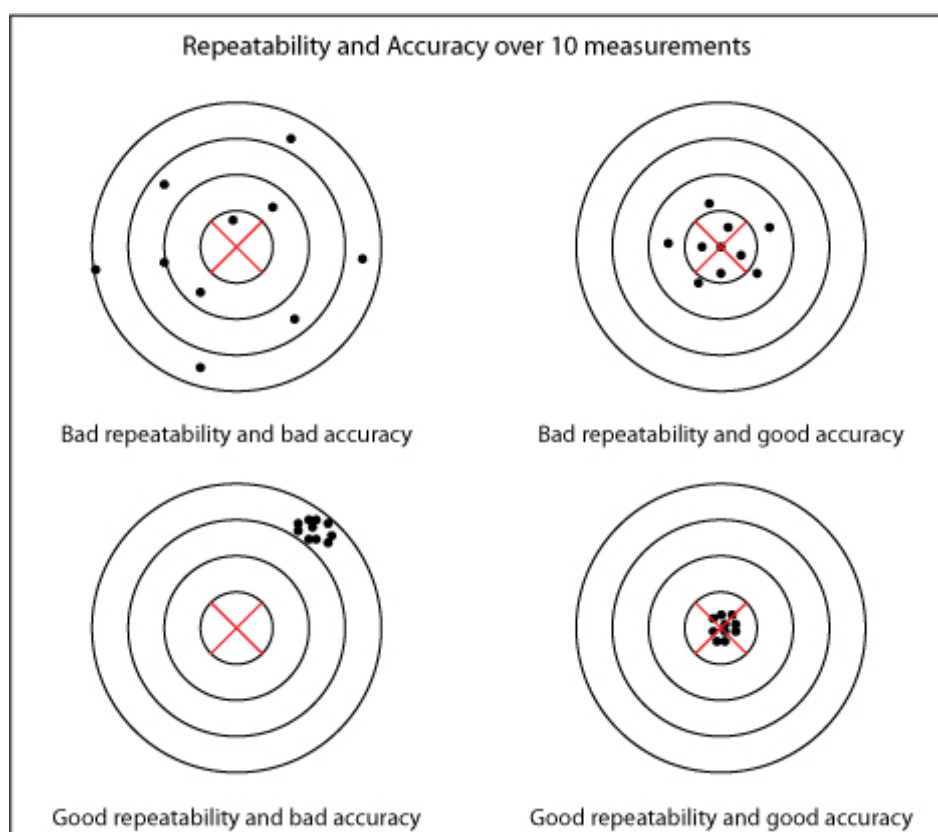


Figura 26 - Acuracidade e Repetibilidade [19].

9.1 Teste de desempenho

Para testar a acuracidade e a repetibilidade do robô foi proposto um teste onde se definiu cinco pontos de teste, o teste foi realizado em cinco etapas, em cada etapa o robô deveria tocar os pontos de teste em uma ordem diferente. O teste foi desenvolvido para testar acuracidade ao medir o erro de posicionamento em cada ponto de teste e concomitantemente medir a repetibilidade visto que cada vez que o manipulador vai ao mesmo ponto teste partindo de origens diferentes. A Tabela 20 mostra a relação dos pontos de teste e a Tabela 21 a ordem dos pontos em cada etapa. O vídeo [20] mostra um exemplo de execução do teste. Devido à falta de outros recursos técnicos os testes foram realizados para o plano base.¹

	PT1	PT2	PT3	PT4	PT5
X	230	140	0	-140	-230
Y	120	240	250	240	120

Tabela 20 – Pontos de teste (mm).

	Etapa 1	Etapa 2	Etapa 3	Etapa 4	Etapa 5
1º	PT1	PT4	PT5	PT3	PT2
2º	PT2	PT3	PT4	PT5	PT1
3º	PT3	PT5	PT1	PT2	PT4
4º	PT4	PT2	PT3	PT1	PT5
5º	PT5	PT1	PT2	PT4	PT3

Tabela 21 - Ordem dos pontos de teste por etapa.

9.1.1 REPETIBILIDADE DE POSICIONAMENTO

É esperado que os resultados de repetibilidade do robô seja muito bom, por se trabalhar com motores de passo que tem apenas alguns pontos de equilíbrio, ou seja, um espaço discreto de variáveis de posição angular e o controle sempre vai colocar no mesmo ponto de equilíbrio (contagem de passos) para uma mesma posição comandada. Neste teste a análise foi qualitativa, pois, como se pode ver nas imagens (onde cada quadrado é de 5x5 mm) a seguir não é possível diferenciar os pontos, assim, configurando uma excelente repetibilidade o que já havia sido reportado no trabalho original [1].



Figura 27 - marcação PT1.

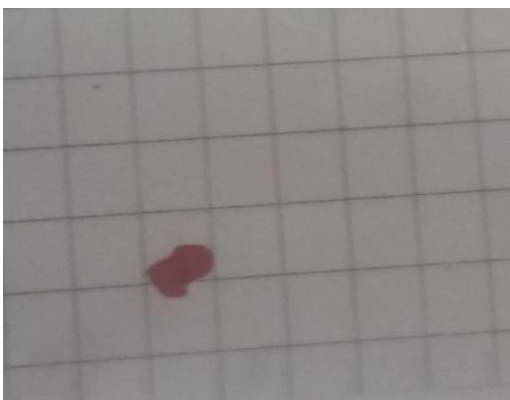


Figura 28 - marcação PT2.

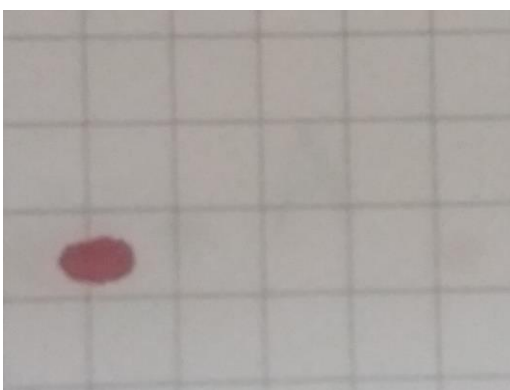


Figura 29 - marcação PT3.



Figura 30 - marcação PT4.



Figura 31 - marcação PT5.

9.1.2 ACURACIDADE DE POSICIONAMENTO

Para determinar a acuracidade foi medido os erros em X e em Y de cada ponto. A Tabela 22 apresenta o resultado das medições e a Tabela 23 os erros absolutos para cada ponto. Por limitações técnicas os erros foram medidos com precisão de 1 mm.

	PT1	PT2	PT3	PT4	PT5
X	232	135	1	-135	-223
Y	120	244	248	237	122

Tabela 22 - Resultado medição pontos de teste (mm).

	PT1	PT2	PT3	PT4	PT5	Média
X	2	5	1	5	7	6
Y	0	4	2	3	2	2.2
Plano	2	6.4031	2.2360	5.8309	7.2801	4.75

Tabela 23 - Erro absoluto dos pontos de teste (mm).

9.1.3 REPETIBILIDADE DE TRAJETÓRIA

Foi realizado um teste qualitativo da repetibilidade de trajetória o robô percorreu cinco vezes a reta com os pontos iniciais e finais descritos em Tabela 24 e o resultado pode ser visto na Figura 32. Pode se notar que em alguns trechos a repetibilidade não foi tão boa quanto a de posicionamento, isto se atribui ao fato de que na Junta 1 o acoplamento permite que haja uma torção entre o eixo do motor e o eixo da Junta, assim, com força exercida do papel na caneta o acoplamento torceu de forma diferente em cada iteração.

	Ponto Inicial	Ponto Final
X (mm)	70	-80
Y (mm)	300	260

Tabela 24 - Reta de teste (mm).



Figura 32 - Resultado teste repetibilidade de trajetória.

10 CONCLUSÃO

Analisando os resultados das modelagens e dos métodos de controle pode-se constatar que mesmo com desvio de projetos esperados de um robô de fabricação não especializada é possível em software diminuir o impacto destes. As formas de controle apresentadas possuem diferentes vantagens e desvantagens, mas podem ser aplicadas conforme se tenha o entendimento de como elas afetam o projeto.

Além dos desvios de projeto o robô tem alguns outros empecilhos como a folga no acoplamento na junta 1 que faz com que a junta não trave na posição desejada. Mas mesmo com essas dificuldades ainda foi possível fazer com que o manipulador tivesse boa repetibilidade para traçar uma reta.

Ainda que simples o sistema de controle por comando em terminal e por leitura de arquivos se mostrou funcional e facilitou a execução dos testes tornando mais eficiente o tempo usado nos testes.

10.1 PROJETOS FUTUROS

Um manipulador desse tipo trabalha com diversos campos da mecatrônica, assim, existe possibilidade de projetos futuros em vários campos temos. Na mecânica:

- Acoplar um redutor ao eixo do motor aumentando a precisão e o torque do robô, além do aumento de precisão pela redução o aumento do torque gera a possibilidade de trabalhar em meio passo aumentando ainda mais a precisão.
- Resolver o problema do acoplamento flexível da junta 1 que permite que o eixo da junta mova com o eixo do motor parado impedindo assim o travamento da junta.

Na área da eletrônica:

- Em vez de controlar com o pwm a tensão nos terminais das bobinas, fazer um controle de corrente permitindo um aproveitamento maior da potência do motor sem por ele em risco, além disso esse sistema resultaria em um sistema mais robusto, protegido de correntes altas.
- Instalar sensores de fim de curso permitindo que o robô vá automaticamente para sua posição inicial e que também sirvam de proteção evitando que o robô tente ir além dos limites físicos de suas juntas.

Para a parte computacional temos:

- Controle de velocidade e aceleração permitindo tanto que o robô tenha velocidade controlável pelo usuário o que pode ser necessário para algumas tarefas quanto permitindo definir um movimento com rampas de aceleração e desaceleração fazendo um movimento mais suave e potencialmente podendo chegar a velocidades mais altas sem perder passo.
- Aprimorar os comandos do robô possibilitando que ele faça mais coisas, ou permitindo que ele aceite uma outra linguagem de comando.

Alguns trabalhos futuros podem ainda envolver mais de uma área de atuação:

- Uma plataforma para instalações de diferentes efetadores terminais que permita o acionamento de efetadores que possam ser controlados como uma garra, por exemplo.

11 REFERÊNCIAS BIBLIOGRÁFICAS

1. MELLO, O. P. M. **Desenvolvimento de um Robô Manipulador SCARA**. Trabalho de graduação em engenharia mecatrônica, Universidade de Brasília. Brasília, p. 118. 2016. (Publicação FT.TG-nº 01/2016).
2. RODRIGUES, D. L. et al. **MANIPULADOR ROBÓTICO HORIZONTAL TIPO SCARA - ROBÔ S4 UMA PLATAFORMA ABERTA PARA DESENVOLVIMENTOS E IMPLEMENTAÇÕES EM ROBÓTICA**. Artigo 4º SBAI - Simpósio Brasileiro de Automação Inteligente, PUC Minas. São Paulo, p. 6. 1999.
3. ARAÚJO, V. B.-H. N.; MEDEIROS, J.; BITENCOURT, A. **Desenvolvimento de robô manipulador com juntas rotacionais e prismática**. Instituto Federal de Educação, Ciência e Tecnologia da Bahia - IFBA. Tocantins, p. 8. 2012.
4. WIKIPEDIA, THE FREE ENCYCLOPEDIA. Stepper motor - Wikipedia. **Wikipedia, the free encyclopedia**. Disponível em: <https://en.wikipedia.org/wiki/Stepper_motor>. Acesso em: 21 Dezembro 2016.
5. STMICROELECTRONICS. L298 - Dual Full Bridge Driver - STMicroelectronics. **Site da STMicroelectronics**. Disponível em: <http://www.st.com/content/st_com/en/products/motor-drivers/brushed-dc-motor-drivers/l298.html>. Acesso em: 15 Novembro 2016.
6. CAMPOS, B. A. N. **Simulador para Modelagem e Calibração de Robôs Industriais**. Dissertação de mestrado em sistemas mecatrônicos, Universidade de Brasília. Brasília, p. 153. 2006.
7. WIKIPEDIA, THE FREE ENCYCLOPEDIA. Denavit-Hartenberg parameters - Wikipedia. **Wikipedia, the free encyclopedia**. Disponível em: <https://en.wikipedia.org/wiki/Denavit%E2%80%93Hartenberg_parameters>. Acesso em: 21 Dezembro` 2016.
8. ROSÁRIO, J. M. **Principios de mecatrônica**. São Paulo: ed: Prentice Hall Brasil, 2005.

9. TUTORIALES. Tutorial de Uso del Módulo L298N. **Naylamp Mechatronics**. Disponível em: http://www.naylampmechatronics.com/blog/11_Tutorial-de-Uso-del-M%C3%B3dulo-L298N.html>. Acesso em: 05 Dezembro 2016.

- 10 FDOMINEC. File:Ttl inside 7404.svg. **Wikimedia**. Disponível em: https://commons.wikimedia.org/wiki/File:Ttl_inside_7404.svg#>. Acesso em: 5 Dezembro 2016.

- 11 ARDUINO. Arduino - GenuinoBrand. **Site do Arduino**. Disponível em: <https://www.arduino.cc/en/Main/GenuinoBrand>>. Acesso em: 15 Novembro 2016.

- 12 PROTOSHIELD Arduino UNO. **HUINFINITO**. ISSN <http://www.huinfinito.com.br/shields-placas-extensveis/477-kit-arduino-protoshield.html>. Acesso em: 05 Dezembro 2016.

- 13 ROCHA, E. Programas Tg Edrysson. **Google Drive**. Disponível em: <https://drive.google.com/drive/folders/0B9NGYYZeINtWMG5pN1JGVjJW0k?usp=sharing>>. Acesso em: 5 Dezembro 2016.

- 14 SIMPLE DIRECTMEDIA LAYER. Simple Directmedia Layer. **Sita da SDL**. Disponível em: <https://www.libsdl.org/>>. Acesso em: 15 Novembro 2016.

- 15 ARDUINO. Arduino Playground - CPPWindows. **Site do Arduino**. Disponível em: <http://playground.arduino.cc/Interfacing/CppWindows>>. Acesso em: 15 Novembro 2016.

- 16 ROCHA, E. Reta. **Youtube**. Disponível em: <https://www.youtube.com/watch?v=1JA-oHS3fFM>>. Acesso em: 04 Dezembro 2016.

- 17 ROCHA, E. Parabola. **Youtube**. Disponível em: <https://www.youtube.com/watch?v=g7YRS2QheIY>>. Acesso em: 04 Dezembro 2016.

- 18 MELLO, M. S. V. **Análise de Desempenho e Melhorias em Robô Manipulador**. Trabalho de graduação em engenharia mecânica, Universidade de Brasília. Brasília, p. 123. 2016.

- 19 BOUCHARD, S. Robotic Gripper Repeatability Definition and Measurement. **Robotiq**. Disponível em: <http://blog.robotiq.com/BID/36551/ROBOTIC-GRIPPER->

REPEATABILITY-DEFINITION-AND-MEASUREMENT>. Acesso em: 04 Dezembro 2016.

20 ROCHA, E. Teste de acuracia e repetibilidade. **Youtube**. Disponivel em: . <<https://www.youtube.com/watch?v=B78xJrVg-eo>>. Acesso em: 04 Dezembro 2016.